

# CA Spectrum®

## Service Manager User Guide

Release 9.4



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2014 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

## CA Technologies Product References

This document references the following CA Technologies products:

- CA Spectrum®
- CA Spectrum® Service Manager (Service Manager)
- CA Spectrum® Report Manager (Report Manager)
- CA Spectrum® Modeling Gateway Toolkit (Modeling Gateway)

## Contact CA Technologies

### Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

### Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to [techpubs@ca.com](mailto:techpubs@ca.com).

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.



# Contents

---

## Chapter 1: Introducing Service Manager 11

About Service Manager .....	11
Services .....	12
Service Policy.....	13
Customers .....	14
SLAs .....	14
Service Health Values.....	14
Service Management Features.....	15
OneClick Licenses and Service Manager Privileges .....	16
Service Manager Installation Considerations.....	16
Plan Service Management Implementation.....	17
Service Manager Utilities .....	18
Open the Service Editor .....	18
Open the Service Policy Editor .....	19
Locating Service Manager Components.....	20
Roll-Up Indications for Service Manager Models.....	22
Service Manager Component Views Outside of the OneClick Console.....	22

## Chapter 2: Creating and Managing Services 25

Service Management Solution Design Guidelines.....	25
Service Model Identification and Creation Guidelines.....	26
Phase I: Build a Service Hierarchy Reflecting Your Business Environment .....	26
Phase II: Add Significant Resource Monitoring Capacity .....	28
Phase III: Ongoing Refinement.....	30
Basic Service Definition .....	31
Define the Role of the Service.....	31
Identify Service Resources .....	32
Select Resource Models .....	32
Specify the Service Policy .....	32
Consider Resource Failure Affects on Service Health .....	33
Refining the Service Definition.....	35
Missing Resource Models .....	35
Resource Models Which Are Not Discrete Enough.....	38
Resource Faults that Shouldn't Impact the Service .....	42
Service Impacting Events That Don't Impact Resources .....	43
Patterns of Resource Faults Which Impact Services .....	44

---

Service Attributes and Relationships .....	47
Create a Service.....	51
Resource Monitors.....	54
Specify the Alarm Types That Affect or Do Not Affect Service Health .....	57
Add a Resource to a Service .....	60
Delete a Resource from a Service .....	61
Edit a Service .....	61
Delete a Service.....	62
Cut a Service .....	62
Service Maintenance Schedule Management.....	63
Create a Maintenance Schedule .....	63
Add a Maintenance Schedule to the Current Schedules List .....	64
Remove a Maintenance Schedule from the Current Schedules List .....	64
Associate an Owner with a Service .....	65
Associate a Customer with a Service.....	65
Service Models in a DSS Environment.....	66
Example: Supported Service to Remote Resource Configurations .....	67

## **Chapter 3: Working with Policies and Policy Components 75**

Policies .....	75
Policy Types .....	76
Create a Policy.....	77
Add Alarm Types to a Custom Condition Policy .....	78
Create a Policy from a Copy .....	79
Edit a Policy .....	79
Delete a Policy.....	80
Attribute Maps .....	81
Create an Attribute Map.....	82
Create an Attribute Map from a Copy .....	84
Edit an Attribute Map .....	84
Delete an Attribute Map.....	85
Rule Sets .....	85
Create a Rule Set.....	87
Create a Rule Set from a Copy .....	88
Edit a Rule Set .....	88
Delete a Rule Set.....	89

## **Chapter 4: Creating and Managing Customers 91**

Customers and Customer Groups .....	91
Create a Customer .....	92
Create a Customer Group .....	93

---

Edit Customer Settings .....	93
Edit a Customer Group .....	94
Move a Customer or a Customer Group .....	94
Delete a Customer or a Customer Group .....	95
Associate a Service or an SLA with a Customer .....	95

## Chapter 5: Creating and Managing Service Level Agreements 97

About Service Level Agreements .....	97
Guarantees .....	98
Period .....	99
SLA Considerations .....	100
Create an SLA .....	101
Create an SLA From an SLA Template .....	103
Guarantee Types .....	105
Create a Guarantee for a Top-Level Service .....	106
Create a Guarantee for a Service, Sub-Service, or Resource Monitor .....	107
Specify Business Hours for a Guarantee .....	109
Edit a Guarantee .....	110
Delete a Guarantee .....	111
Create an SLA Period .....	111
Edit an SLA .....	112
Delete an SLA .....	113
Associate a Customer with an SLA .....	113
SLA Templates .....	114
Create an SLA Template .....	114
Edit an SLA Template .....	115
Delete an SLA Template .....	116
Guarantee Templates .....	116
Create a Guarantee Template .....	116
Edit a Guarantee Template .....	117
Delete a Guarantee Template .....	118

## Chapter 6: Creating Service Management Components with Modeling Gateway 119

About the XML Framework .....	120
Service Models .....	121
Policies and Watched Attributes .....	122
Example: Services That Monitor Resources Directly .....	123
Example: Services That Monitor Resources in Resource Monitors .....	124
Example: Using XML to Define a Service Template .....	126
Example: Define a Service Maintenance Schedule .....	133

---

Example: Define an Alarm Exemption List for a Service or Resource Monitor .....	133
Example: Associate an SLA to a Service.....	135
Examples: Create a Guarantee for an SLA.....	135
Example: Define an SLA .....	136
Example: Define a Customer and a Customer Group .....	137
Example: Import XML Input Files .....	140
Service Attributes (SM_Service) .....	141
Monitor Resource Monitor Attributes (SM_AttrMonitor) .....	142
Customer Group Attributes (SM_CustomerGroup) .....	143
Customer Attributes (SM_Customer).....	143
SLA Attributes (SM_SLA) .....	144
Guarantee Attributes (SM_Guarantee).....	145
Schedule Attributes (Schedule) .....	147

## **Chapter 7: Monitoring Service Management Components with the Service Dashboard 149**

The Service Dashboard.....	149
Open the Service Dashboard.....	151
Topology and List Views in the Contents Panel.....	153
Explorer Folders and Topology Icons .....	153
Status Indicators.....	154
Access Information about a Service Management Component.....	155
Service Dashboard Interface Management.....	157
Locate Service Management Components .....	158
Print Dashboard Views .....	159
Export Dashboard Views .....	160
Use the Service Dashboard Editing Tools .....	160
Service Outage Management.....	162
View Current Outages .....	162
View Outage History .....	163
Service Outages.....	163

## **Chapter 8: Monitoring Service Management Components with Unicenter Management Portal 167**

About the Service Level Manager Portlet .....	167
Publish the Service Level Manager Portlet in UMP.....	168
View Service Information .....	169
View SLA Information.....	170
View Customer Information .....	170
Open the OneClick Console and the Service Dashboard.....	171



---

Apply and Manage Layouts .....	171
--------------------------------	-----

## **Chapter 9: Generating Service Manager Reports 173**

Service and SLA Reports .....	173
Outage Reports .....	175
Inventory Reports.....	176
Detailed Availability Reports .....	177
Summarized Availability Reports .....	177
Customer Reports .....	178
SLA Status Reports .....	179
Health Reports .....	181
Generate Reports .....	182

## **Appendix A: Service Manager Policy Descriptions 183**

Policy ID Mappings .....	183
Condition Value Sum Greater Than Or Equal .....	185
Port Status Policies .....	186
Condition Policies .....	187
Response Time Policies .....	188
Service Health Policies .....	189
Contact Status Policies .....	190

## **Appendix B: Resource Monitor Implementation 193**

Policy Implementation: Monitor Routers .....	193
Resource Monitor Implementation: Monitor Routers and Their Ports .....	194
Refined Resource Monitor Implementation: Monitor Routers, Ports, and Response Time Tests .....	195

## **Appendix C: Administration and Maintenance 197**

Customize a Service Editor Information Table .....	197
Customize a Service Policy Editor Information Table .....	198
Remove Service Manager Historical Data from All Landscapes .....	198
Remove Service Manager Historical Data from a Single Landscape .....	199
Remove Destroyed Service Manager Models from All Landscapes .....	199
Custom Resources Table .....	200
Create the Custom Table File .....	201
Example: Resources Table Configuration File .....	201

## **Index 205**



# Chapter 1: Introducing Service Manager

---

This section contains the following topics:

[About Service Manager](#) (see page 11)

[Service Management Features](#) (see page 15)

[OneClick Licenses and Service Manager Privileges](#) (see page 16)

[Service Manager Installation Considerations](#) (see page 16)

[Plan Service Management Implementation](#) (see page 17)

[Service Manager Utilities](#) (see page 18)

[Locating Service Manager Components](#) (see page 20)

[Roll-Up Indications for Service Manager Models](#) (see page 22)

[Service Manager Component Views Outside of the OneClick Console](#) (see page 22)

## About Service Manager

CA Spectrum Service Manager is a tool that provides the capability to monitor and manage IT infrastructure that is based on the business services that it provides. Rather than managing a collection of network devices, servers, and applications; you can organize and manage these elements based on how they provide or support specific services. CA Spectrum service models offer visibility on how the infrastructure elements affect the availability of the business services. This visibility aids in prioritizing infrastructure faults that are based on their impact to business services, and highlighting weaknesses in the environment.

The Service Manager application includes a comprehensive set of tools for creating, managing, and monitoring business services, Service Level Agreements (SLAs), and service customer models in CA Spectrum. Leveraging CA Spectrum fault-management capabilities, Service Manager provides real time and historical insight into the status of your service management components. It also provides a suite of reports for all service management components which can be generated with the CA Spectrum Report Manager application.

You can manage and monitor Service Manager components in the following interfaces:

- The OneClick Console provides administrative personnel complete access to Service Manager configuration editors and service management models.
- The Service Dashboard provides service providers and customers access to status views of service management models and service outage management tools.
- The Service Level Manager portlet, which can be incorporated into the Unicenter Management Portal (UMP), provides summary status information about services, SLAs, and customers to Unicenter users with security access to Service Manager models.

Service Manager lets you extend your infrastructure management capabilities beyond the per-device and per-application level. It provides you with the tools to build mechanisms that let IT-service providers and customers validate service availability and performance.

## Services

A CA Spectrum service is a model that represents some logical business service. For example, a router model represents the status of physical device similarly, a service model represents status of the business service.

A service model contains a set of resources and a policy indicating the behavior of resources. Service resources are other CA Spectrum models which collectively provide or support the availability of a business service. A service is only available when its resources are available. In turn the service model monitors the availability of its resources to determine its health or availability. By applying the collective resource availability to its service policy, the service model can depict the real-time health of the business service it represents.

A service model differs from other types of models in CA Spectrum. To confirm this information, perform the following actions:

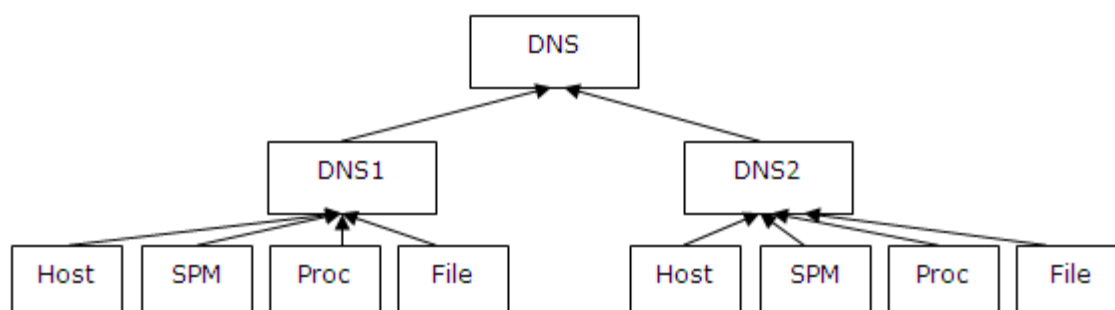
- Consider some of the patterns that are common in IT management. For example, redundant devices, hot swap back ups, load balanced servers, clustered or pooled resources.
- Verify how these patterns are used to support specific business services. One of the oldest and most common business services is DNS. The critical nature of DNS lets you deploy multiple DNS servers. DNS improves performance and also mitigates the risk of DNS failure.
- Represent the real-time status of the DNS service in an environment with two servers that are dedicated to DNS. Monitor the servers themselves as host models in CA Spectrum. The host model lets you understand the situation if contact is lost to the server and provides information about a number of other potential system faults. You can create SPM tests to validate that each server is responsive. You can also monitor critical processes or file systems on the servers.

### Availability of DNS

DNS is available when both servers are functioning. If one of the servers is down, DNS is still available. However, DNS is not available when both the servers are down. The host models, SPM tests, monitored processes, and file systems are all individual models, any of which can fail and produce an alarm. You can view alarms to determine the availability of DNS which can become complex and error prone with this small set of models.

You can manage this complexity using CA Spectrum Service Manager. Using service models, you can organize the host models, SPM tests, processes, and file systems to represent the DNS service and express the availability of that service.

The following diagram illustrates the process to implement DNS in CA Spectrum Service Manager.



## Service Policy

In addition to the resources that constitute a service, each service specifies a policy that infers a service viability (its service health) based on the collective status of its resources. The role of the service policy is to create an accurate representation of the service health that is based on health of the resources that comprise the service. The objective is to ensure that the health indicated by the service model in CA Spectrum reflects the performance of service. Selecting or creating the appropriate service policy is essential to ensure that service health is accurately represented in CA Spectrum.

A policy specifies a single attribute to monitor for the collective set of service resources. The policy is comprised of two basic components, the Attribute Map and the Rule Set. The Attribute Map associates resource attribute values to service health values. The Rule Set defines the health of the service by giving a set of resource health values.

A service can also include one or more resource monitors instead of a single policy. Each resource monitor can use its own policy extending the monitoring capability of the service.

A number of policies are shipped with CA Spectrum Service Manager which represent common resource monitoring patterns. If applicable, you can use these policies to familiarize yourself with the creation of custom policies. The use of custom policies can be necessary to verify the accuracy of health that is represented by the service model.

## Customers

A CA Spectrum Customer model or Customer represents any person or an organization that uses services or is a party to a Service Level Agreement (SLA). CA Spectrum Service Manager lets you associate customers with the services and SLAs to monitor the service management components on a per-customer basis.

## SLAs

A CA Spectrum SLA model, or *SLA*, comprises one or more service *guarantees* that specify the service obligations stipulated in an SLA contract for a particular time period (for example, week, month). Service Manager lets you specify the following two types of guarantees:

- Availability
- Response time.

Both types of guarantees record service outage time and compare it against a user specified threshold for a period. Availability guarantees also support supplemental thresholds (Mean Time to Repair, Mean Time Between Failure, and Maximum Outage Time).

Create SLAs from scratch or from SLA templates and associate multiple SLAs with a single service and multiple customers with an SLA.

## Service Health Values

A *service health* value indicates the viability of a service, whether it is operating at an acceptable or less than acceptable level or is inoperable. Service Health also indicates whether a service is in maintenance mode or in an initial state, if the service has no resources.

The following table lists and describes service health values and corresponding CA Spectrum alarm states.

Service Health	Description	Icon Color
Up	The service is operating normally.	Green
Down	The service is unavailable. The service is experiencing a critical outage.	Red
Degraded	The service is available but operating at a limited capacity. The service is experiencing a major outage.	Orange

Service Health	Description	Icon Color
Slightly Degraded	The service is available but operating at a slightly diminished capacity. The service is experiencing a minor outage.	Yellow
Maintenance	The service has been put into maintenance mode and is not actively monitoring resources. The service is experiencing a maintenance outage.	Brown
Loss Of Management	The SpectroSERVER has been shut down. The service is experiencing a loss of management outage.	Gray
Defunct	The service has a configuration error. The service stays in the nonfunctional state until the error has been corrected.	Blue
Initial	The service has no resources associated with it. The service stays in an initial state until resources have been associated with the service.	Blue

## Service Management Features

CA Spectrum Service Manager includes the following features:

- Views that let you monitor the health of services in real time and relate the services to customers affected by IT infrastructure faults.
- Service health records that provide the basis for reports that you can generate on a scheduled or on an on-demand basis using Report Manager.
- SLA violation alarms that notify you when an agreement has been violated or is in danger of being violated.
- Root cause analysis of any service degradation (in terms of infrastructure alarms).
- The capability to designate maintenance periods for services.
- The capability to exempt any service outage from impacting an SLA.
- Extensions to Modeling Gateway that let you create service management components and schedule service maintenance through an XML feed.
- A web page that is added to UMP for viewing service management-related information.

## OneClick Licenses and Service Manager Privileges

To access Service Manager from the OneClick Console, OneClick administrator privileges or operator license is required. To access the Service Dashboard, Service Manager license is required. For more information, see the *CA Spectrum Administrator Guide*.

The following table compares license types and the default roles and privileges that are associated with them:

OneClick License	Default Role	Default Privileges
Operator	OperatorRW	Ability to update service descriptions and view service information and the Service Management hierarchy in the OneClick Console.
Administrator	AdministratorRW	Ability to create and edit services, customers, and SLAs with Service Editor, and create and edit policies, attribute maps, and rule sets with Service Policy Editor.
Service Manager	ServiceManagerRW	Access Service Dashboard.

**More information:**

[The Service Dashboard](#) (see page 149)

## Service Manager Installation Considerations

**Important!** Service Manager must be installed on both the SpectroSERVER and OneClick servers.

Review the following Service Manager installation considerations:

- The modeling catalog and all modeling intelligence exist within SpectroSERVER.
- The historical database and event handling code exist on the OneClick web server which is installed with OneClick.
- All the client UI components and dashboard are installed with OneClick.
- If you have a separate OneClick installation for Report Manager, install Service Manager on that server to populate the Service and SLA reporting tables.



## Plan Service Management Implementation

To benefit fully from the Service Manager capabilities, plan your service management implementation. You can contemplate the following questions and considerations:

- What business services do you want to monitor?
- What particular resources — processes, software applications, and IT devices — support those services?
- How can conditions and faults that affect services be detected? Which resource attribute(s) can be monitored to determine the health of a service?
- Who can be notified if a service fails?
- What are the service performance obligations, and how can they be quantified?
- What is the criticality of a given service relative to other services?
- Consider implementing a Service Manager solution as an iterative process. In the initial phase, concentrate on obvious resources and obvious faults. Answer the question: the service does not work, if \_\_\_\_\_ is down. This information can give you a good foundation for enhancing the service later.
- Identify common sub services or foundation services. If all the multiple services rely on a common set of resources group (resources in their own service), then you can make that service a resource of higher-level services that depend on it.
- Create empty services if you know that a service exists, but you are not familiar with the resources that comprise it. These services left initial are important. These services represent areas of infrastructure that are not familiar, and need to be explicitly monitored.
- Build your services such that they can be easily enhanced. Rather than services which directly monitor resource models such as devices and applications, build services using resource monitors. It is easy to enhance these services in the future by adding new resource monitors without having to change the overall structure of the service.

## Service Manager Utilities

You can work with the following utilities to create and manage the Service Manager components:

### Service Editor

Lets you create and manage services, SLAs, SLA guarantees, SLA templates, SLA guarantee templates, customers, and customer groups.

### Service Policy Editor

Lets you create and manage service monitoring policies and the constitute components, attribute maps, and rule sets.

### Condition Correlation Editor

Lets you build correlation domains to monitor resources for specific resource events or combinations of events. These correlation domains can then be used as service resources, extending the monitoring capability of the service from attribute monitoring to event monitoring.

## Open the Service Editor

The Service Editor is the primary administrative tool for configuring service models, SLAs, and Customers. The Service Editor displays service models in a flat list or in a hierarchy view. From the Service Editor, you can create new service models or can edit existing service models.

### Follow these steps:

1. Select the OneClick Console or the Service Dashboard.
2. Click Tools, Utilities, Service Editor, from the main menu.

**Note:** The Service Editor is not available if you do not have access privileges or the OneClick installation does not include the Service Manager product.

The Service Editor opens.

## Open the Service Policy Editor

The Service Policy Editor lists service monitoring policies and policy specifications and includes commands for creating and managing policies. It lets you create and manage policies and their constituent components, attribute maps, and rule sets.

**Follow these steps:**

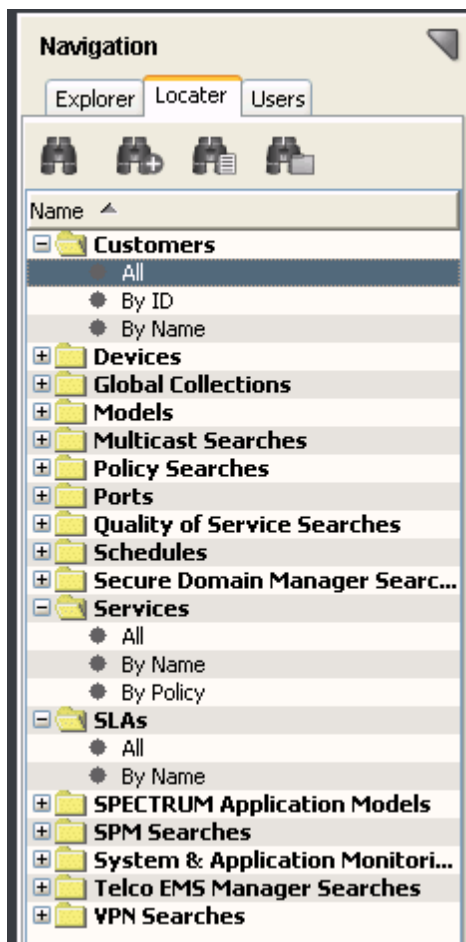
1. Select the OneClick Console or the Service Dashboard.
2. From the main menu, click Tools, Utilities, and Service Policy Editor.

**Note:** The Service Policy Editor is not available if you do not have access privileges or the OneClick installation does not include the Service Manager product.

The Service Policy Editor appears.

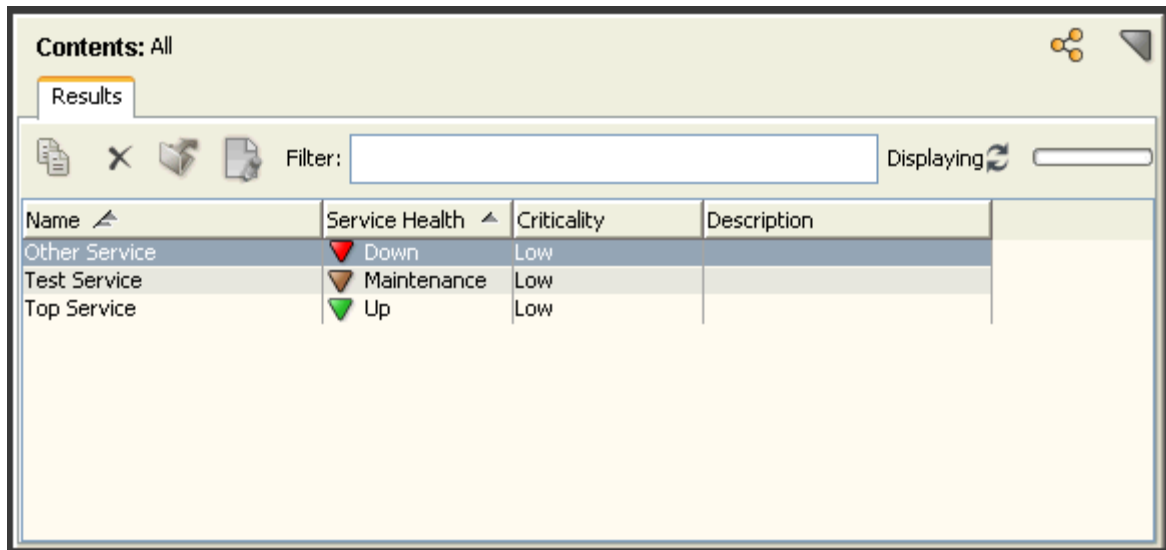
## Locating Service Manager Components

You can locate and display services, service customers, and SLAs in the Locator tab of OneClick Console Navigation panel. Search can be initiated from Customers, Services, and SLAs folders, as shown in the following image:



You can track the Service Manager components in a distributed SpectroSERVER environment. For more information, see the *CA Spectrum Operator Guide*.

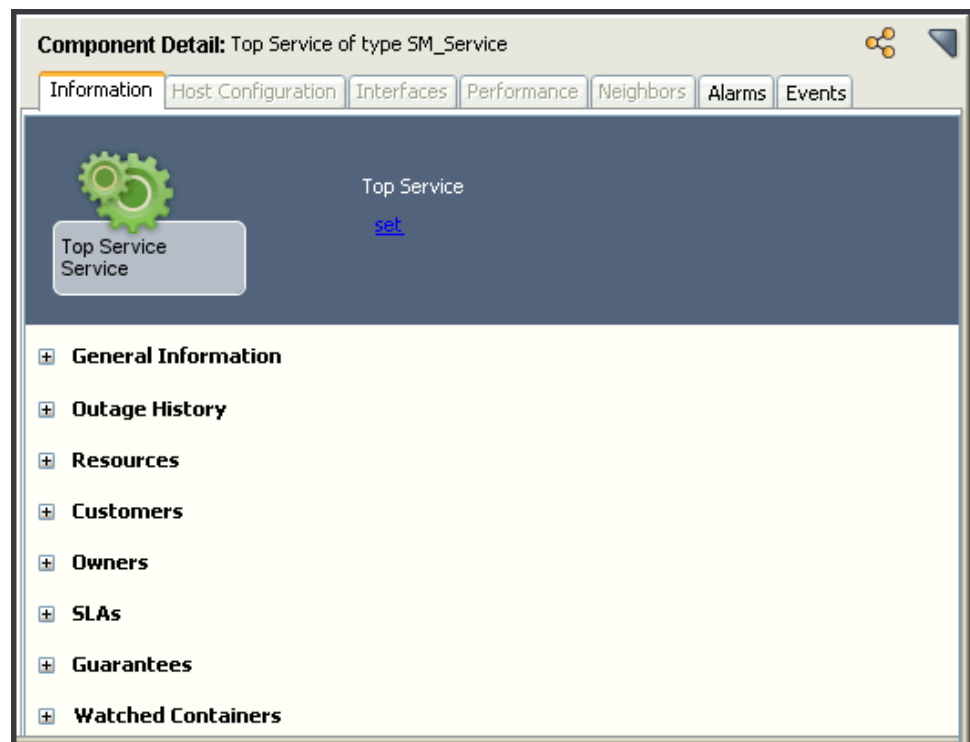
The following image displays the search results that appear in the Contents panel of OneClick Console:



The screenshot shows the 'Contents: All' panel with a 'Results' tab selected. It features a search bar and a table of service components. The table has columns for Name, Service Health, Criticality, and Description. The 'Other Service' row is highlighted.

Name	Service Health	Criticality	Description
Other Service	Down	Low	
Test Service	Maintenance	Low	
Top Service	Up	Low	

The following image displays the details about all aspects of the selected entry in the Component Details panel:



The screenshot shows the 'Component Detail' panel for 'Top Service of type SM\_Service'. It includes tabs for Information, Host Configuration, Interfaces, Performance, Neighbors, Alarms, and Events. The 'Information' tab is active, displaying a gear icon, the service name 'Top Service', and a 'set' link. Below this is a list of expandable sections: General Information, Outage History, Resources, Customers, Owners, SLAs, Guarantees, and Watched Containers.

**Component Detail: Top Service of type SM\_Service**

Information Host Configuration Interfaces Performance Neighbors Alarms Events

Top Service  
[set](#)

- + General Information
- + Outage History
- + Resources
- + Customers
- + Owners
- + SLAs
- + Guarantees
- + Watched Containers

## Roll-Up Indications for Service Manager Models

The Service Management tree consists of a set of four models. The root model is the Service Management model, which contains three top-level manager models, such as Service Manager, Customer Manager, and SLA Manager models. These top-level managers organize all of the user created service management components. All service models can be organized under the Service Manager model which is displayed in OneClick with the name Services. Similarly all SLA models and customer models appear under SLAs and Customers.

The Condition of each of these manager models is equivalent to the worst status of all contained models. For example, if the worst service within the landscape has a service health of Degraded, the Service Manager model on that landscape has a Condition of Major. If the worst SLA within the landscape has a SLA status of Violated, the SLA Manager has a Condition of Critical. You can view the Condition of a manager model by the icon color. No alarms are associated to the Condition of the manager models.

Within the Service Dashboard, all landscapes are merged into a single tree in the Explorer panel. Within the service dashboard, a set of folders represent the collective set of manager models from each landscape. Regardless of how many landscapes occur within a DSS the Service Dashboard displays only one folder for each category of manager model. These folders have the same names as the set of models within the SpectroSERVER: Services, Customers SLAs. The icon for each folder displays the status of the worst top-level model within the DSS. If the worst Service Manager status within the DSS is Minor, the Service Folder in the Service Dashboard displays a yellow icon. Similarly, if a Customer Manager within the DSS has a Condition of critical, the Customers folder in the Service Dashboard displays a red icon.

## Service Manager Component Views Outside of the OneClick Console

You can view the Service Manager components outside of the OneClick Console from the following interfaces:

### **Service Dashboard**

The Service Dashboard is a service management only view. It provides visibility to the real-time status for all Services, Customers, and SLAs. The Service Dashboard provides OneClick like console with an Explorer tree, component detail view, and topology view. In addition, the Service Dashboard provides some historical service management views including service outage history, and SLA/Guarantee trending. The dashboard is designed for anyone who is interested in monitoring Service Manager components, and enforces CA Spectrum model security.

### **Unicenter Management Portal**

CA Unicenter users can view summary information about Service Manager components by adding content from the Service Manager portlet. The Service Manager portlet displays flat lists of service models, customer models, and SLAs models. The Service Manager portlet also provides context launching to the OneClick Console or Service Dashboard.





# Chapter 2: Creating and Managing Services

---

This section contains the following topics:

[Service Management Solution Design Guidelines](#) (see page 25)

[Service Model Identification and Creation Guidelines](#) (see page 26)

[Basic Service Definition](#) (see page 31)

[Refining the Service Definition](#) (see page 35)

[Service Attributes and Relationships](#) (see page 47)

[Create a Service](#) (see page 51)

[Add a Resource to a Service](#) (see page 60)

[Delete a Resource from a Service](#) (see page 61)

[Edit a Service](#) (see page 61)

[Delete a Service](#) (see page 62)

[Cut a Service](#) (see page 62)

[Service Maintenance Schedule Management](#) (see page 63)

[Associate an Owner with a Service](#) (see page 65)

[Associate a Customer with a Service](#) (see page 65)

[Service Models in a DSS Environment](#) (see page 66)

## Service Management Solution Design Guidelines

The key to a successful service management deployment is accuracy in service modeling. The goal when designing services can be to create the most accurate representation possible. Given that objective, often you do not understand how each service functions, and all of the resources on which the service relies. One of the most challenging aspects of building a service management environment is collecting all of the information that is required to model each service. For this task, you require a significant amount of collaboration to create accurate services.

It is rare that a single individual has all the information that is required to build a large service management environment. In most businesses, different teams specialize in different areas of the business each focused purely on their own domain. Ironically it is this typical business structure that creates the need for a service management solution. The more difficult it is to build the service management environment, the greater the need for doing so.

The following section outlines some guidelines for planning and building a service management solution. It takes time to plan and to model a large and complex service management environment. Investing the time to collect information and develop a plan for building the service management solution helps to confirm your success. Part of your plan can include incremental goal, identifying a timeline and milestones for the project. A representation of services for each organization can be included, perhaps a demonstration of how specific infrastructure faults are now processed and prioritized based on service impact. Establishing these milestones lets you realize value throughout the process, and motivates you to expand your solution continually.

One final note on planning is to recognize that you are building a service management solution that must continue to evolve. Your plan may identify a time and point of completion, but that does not mean you will never again have to create or edit a service model. Business infrastructure is continually changing, and you may find that services change too. Your goal is to build a service management solution that is flexible and easy to maintain. On performing this task, you can confirm that your solution can accommodate the dynamic nature of your business.

## Service Model Identification and Creation Guidelines

Determining where to start creating your service management environment is the most daunting task. Various techniques such as top down, bottom up, most critical first are available. Regardless of which technique you select, follow the available guidelines.

The first step is to define a phased approach. Develop a multi-phase plan, before you create any service models. The goal of this plan to two folds. A multi-phase plan helps you manage the complexity of creating a large environment and confirms that you receive incremental value throughout the deployment process. It is important that you are able to recognize value from each service that you create and so from each phase of your deployment.

Each environment is different, but the following three-phase deployment plans can work for you, or can be adapted to fit your environment.

### Phase I: Build a Service Hierarchy Reflecting Your Business Environment

Phase I focuses on building all of the major services, and establishing the service hierarchy which reflects your business environment. This phase produces all of the high-level business-related services, and many of the lower layer services on which they rely. Phase I implementation does not provide a comprehensive solution that encompasses all possible service faults. But creates a hierarchy which shows the services, each organization relies upon and the impact those services have on the overall business.

Throughout this process, identify services with which you are not familiar and for which you cannot determine resources. You can create an empty service. Defining empty services can shed light on areas of the business or infrastructure which are not familiar.

Generally you can start identifying high-level services by listing different organizations within your business environment. If your business is a service provider, this may include different customers and core service components that are shared by those customers. If you are working in an enterprise environment often, you can simply list the different departments within the business. These departments can frequently be considered high-level services. You can also identify dependencies within these high-level services.

Once you have determined the high-level business-specific services, think about what services they rely on. This next tier of services is commonly associated to specific job functions within an organization. These services may be related to specific applications, specific data, and specific security access. Identify various specialized and shared services.

Once you have identified the services upon which various organizations rely, consider the requirements for providing those services. At this level, identify servers for applications or databases. Depending of the nature of these services, recognize specialized service hierarchies and application interdependencies.

You have now defined a broad set of services which are critical to various job functions. This lets you look for shared component services that these job function services require. Regardless of the variety of applications or access areas, you can find many common services that provide the foundation for the specialize services. At this level, you see common networking or domain management services like DNS and DHCP.

Finally, determine the shared network resources upon which all higher-level services rely. Locate core areas of networks that manage traffic and control access. These core services ultimately impact virtually all other services which you have identified.

You have probably collected enough information to start creating service models. For this initial phase of service creation focus on the most obvious resources and obvious faults. The goal of this phase is not to create comprehensive services, but to establish the broad service hierarchy which depicts how the business functions.

It is easy to fall into the trap of trying to define a perfect service that covers all possible resource faults. The problem is that you quickly get mired by complexity, and require assistance from domain experts who understand the intricacies of each service. This process consumes much time, and can stall the process of building the overall service hierarchy. You can revisit each service in subsequent phases and can refine the resource monitoring capacity to handle the more complex fault scenarios.

As a guideline for determining resources for the first phase, complete the following statement: "The service won't work if \_\_\_\_\_ is down". The answers to this question are the resources that you can monitor for the first phase. Take care not to dwell on all of the potential resources. Create the service with the obvious resources and move on. Most of your services for phase I have other services as their resources. You can determine that one service relies on another service. Although you do not understand how the other service works, you can create the empty service and move on.

Phase I is complete when you have built a service hierarchy which reflects your business environment. Consult with others and validate your overall design. The hierarchy that you have created serves as the baseline for subsequent phases.

## Phase II: Add Significant Resource Monitoring Capacity

The goal of phase II is to increase the accuracy of your service models. You can expect the following two things on increasing accuracy:

1. Adding resource monitoring capacity to encompass more potential faults, and eliminating resource monitoring configuration which may produce false service outages.
2. Enhancing your resource monitoring capacity such that you can identify various levels of service degradation instead of simply showing that a service is Up or Down.

Typically in phase II you can revisit the lower layer services which you created in phase I and any empty services that are identified in phase I.

For phase II, you can expand many of the lower layer services by adding new resource monitoring capacity. Refine the resource statement to include the following subtle faults:

- The service won't work if \_\_\_\_\_ is not available
- The service won't work if \_\_\_\_\_ is not responding
- The service won't work if \_\_\_\_\_ is not running
- The service won't work if \_\_\_\_\_ is not found
- The service won't work if \_\_\_\_\_ is slow
- The service won't work if \_\_\_\_\_ is at maximum capacity

Consider the following statements to identify resource faults which degrade service health:

- The service will be slow if \_\_\_\_\_ is down
- The service may not work if \_\_\_\_\_ is not available

- The service cannot perform these functions if \_\_\_\_\_ is not accessible
- The service will not be able to handle all requests if \_\_\_\_\_ is at maximum capacity

In phase II, you can start adding monitoring capability for system resources, performance statistics, and response times. You can refine the existing service resources by identifying more discrete resources, such as the specific interfaces that a service relies instead of devices.

In addition to monitoring, the Condition or status of resource models you can add new resource monitors to evaluate the value of other attributes which express the performance of a model.

Throughout phase II, identify many new service models which encompass subtle, but critical pieces of functionality upon which other services rely.

Look for dependencies between services particularly where an application server relies on another application. Look for areas where network configuration is critical to data processing for security access or quality of service.

Once again take care not to go too deeply into the intricacies of each service. Limit your phase II enhancements to the type of refinements common to many services. For example, regardless of the application, system resources affect availability. Additionally, poor response time between various parts of network impacts many higher-level services.

In particular, do not yet expand resource monitoring to include application-specific detail. This type of monitoring can be addressed in Phase III.

Phase II is complete, when you have added significant resource monitoring capacity. Except for specific application faults, your solution should now be able to process a much wider set of enterprise faults. Similarly, your solution should also be able to report various levels of service degradation.

## Phase III: Ongoing Refinement

Phase III can be considered the ongoing phase. Regardless of how complete the solution seems to be, there is most likely some potential to refine it further. During phase III, you can extend your resource monitoring capacity to the most complex resource faults. It is difficult to define an end point for phase III, but one guideline is to consider the resolution process. If the resolution process is the same regardless of the fault your monitoring capacity is granular enough. It is tempting to add new resource monitors continually for specific types of faults, but this task is only useful if the course of action differs for each fault.

Depending on your level of expertise, it is the phase where you are most likely to require the collaboration of other domain experts for specific services.

When dealing with application-based services, collect information about the application-specific resources that can be monitored. For example, specific processes, files systems, and log files. Add new services and resource monitors for these various aspects of the application. You can also add monitoring for specific connection ports, and where possible specific transactions.

As part of phase III, add monitoring capacity for configuration changes. It includes monitoring of network configuration policies, and server configuration utilities. Look for configuration changes that can degrade or otherwise impact services.

Whenever possible include monitoring of user access and security mechanism for different applications or networks.

Look for scenarios where a service is impacted even if no specific fault occurs on the service resources. Also look for instances where a service is impacted differently depending upon a combination of resource faults.

You can again refine the following service resource statements to look for additional types of faults:

- The service won't work if the \_\_\_\_\_ process is not running.
- The service won't work if \_\_\_\_\_ is restricting access.
- The service won't work if the \_\_\_\_\_ queue is full.
- The service won't work if the \_\_\_\_\_ is archiving.
- The service won't work if \_\_\_\_\_ doesn't authenticate the user.

Each time that a new resource monitoring capacity to a service you are expanding the set of resource faults the service covers, increases the accuracy of the service.

It is difficult to define when phase III is complete. New resource faults are discovered which were not previously identified. Ideally your design can accommodate these new faults. Each time a new fault is discovered, adapt your resource monitoring such that the next time the fault occurs you can correctly understand its service impact.

## Basic Service Definition

Once you have determined what services to model, you can identify several key properties of each service and can characterize the behavior of the service under certain circumstances. To complete this process, perform the following tasks:

- [Define the Role of the Service](#) (see page 31)
- [Identify Service Resources](#) (see page 32)
- [Select Resource Models](#) (see page 32)
- [Specify the Service Policy](#) (see page 32)
- [Consider resource failure affects on service health](#) (see page 33)

### Define the Role of the Service

Primarily, identify the role of the service model. The service role encompasses the capability and purpose of the service.

The capability identifies what the service is monitoring. Each service model must represent a capability that can be monitored. This capability can be a tangible process, such as an email service. Alternately, the monitored capability can be more abstract or can represent human resources service. Regardless of the service role, you can state that the service measures the health of some capability.

Once you have identified the capability that is being monitored, define the purpose for monitoring that capability. The purpose is a statement of value that you expect to get from monitoring the capability. For example, many services are intended to provide real-time fault analysis. You can create a service to know the health of the email system. Or a service that shows the impact of resource faults within your infrastructure. Other services are designed to monitor responsibility or compliance. For example, a service that monitors server availability to confirm that it is in compliance with some specified guarantee. In addition, some service models are used to represent organizational dependencies, such as regional offices, depending on resources from a centralized location.

Determining the role of the service helps you select service resources. You can specify service criticality and can define service relationships with customers and SLAs.

## Identify Service Resources

Once the service role is defined, you can identify service resources. Service resources provide the capability that the service is monitoring.

For example, consider a service with the role of monitoring the real-time availability of a web-based application. Most likely the application is hosted by one or more servers. These servers can be considered as service resources.

It is important to understand that identifying service resources is an iterative process. It is recommended first to identify the most obvious resources that provide the services capability. In the previous example, the servers hosting the web application are the most obvious resources.

Later, determine the process to monitor the services resources. In many cases, determining the process to monitor one resource reveals other previously unidentified resources. You can manage the service modeling process and can implement your solution in phases. The first phase encompasses the most obvious resources, and higher-level faults. Subsequent phases add additional or more discrete monitoring capability.

## Select Resource Models

Once you have identified a service most obvious resources, determine what CA Spectrum models best represent those resources.

Using the example of a service with the role of monitoring, the real-time availability of a web-based application; you can start with device models that represent the servers. These are likely to be Host models of some sort in CA Spectrum, or can simply be Pingable models.

## Specify the Service Policy

Having located the set of CA Spectrum models which represent the service resources, determine the best way to monitor these models. This is the process of specifying the service policy that the service applies to its monitored resources.

The first part of specifying the service policy is identifying the model attribute that can be monitored to determine the status of the resource.



Selecting the attribute to monitor has a lot to do with the type of resource that is being monitored. If the model is a Pingable monitoring its Contact Status attribute can be reliable. If the resource model is a Port model, monitoring its Port Status can be a good choice. Perhaps the resources of the service are other services. In that case, you can monitor the Service Health of the resources models.

For the first phase of service creation, stick to the obvious resources and obvious resource faults. You can identify several attributes that can be used to express the status of a resource model. For the first phase, select the attribute that provides the broadest representation. You can refine the service later by adding more attribute monitoring, or narrowing the specific faults that are associated to an attribute.

Most of the models in CA Spectrum have the Condition attribute. This attribute is commonly associated to alarms on the model. For example, if a model has a major or orange alarm, the value of its Condition attribute is Major. Condition is often the simplest attribute to use for monitoring a resource. If you cannot identify another attribute which expresses a models status Condition is probably a good starting point.

**Note:** It is often easiest when creating services to start by monitoring the Condition of the service resource models. Although, this can be a good starting point, the value of Condition is often influenced by many different types of outages some of which may not be appropriate for the service.

## Consider Resource Failure Affects on Service Health

The next part of specifying the service policy is to consider how each individual resource impact the service. More specifically, consider how the failure of each resource affects the health of the service.

The health of a service model can be expressed by four values, such as Up, Down, Degraded, and Slightly Degraded. Consider each resource that has been identified, and the health of the service should indicate, if that resource fails.

A fault matrix table can be a useful tool to document how various resource failures can impact the health of a service. The fault matrix is a table with columns for each resource, and a column for service health. Here is an example of a simple fault matrix for a pair of servers that support a web-based application. The columns Server 1 and Server 2 contain potential status for each server, the Service Health column contains the logical service health given the status of each Server.

Server 1	Server 2	Service Health
Up	Up	Up
Down	Up	Slightly Degraded
Up	Down	Slightly Degraded

Server 1	Server 2	Service Health
Down	Down	Down

After reviewing the table, you can describe the behavior of the service with the following statements:

- If either Server 1 or Server 2 are Down the Service Health is Slightly Degraded
- If both Server 1 and Server 2 are Down the Service Health is Down

Stated in a resource independent manner, the following two expressions can be identified:

- When any 1 resource is Down the service is Slightly Degraded
- When all resources are Down the service is Down

In terms of a CA Spectrum service policy, these two expressions can be considered rules, and collectively they make up a rule set. The combination of an attribute and a rule set is the service policy.

By first identifying which resource model attribute represents its status, and next identifying a set of rules which describes how each resource impacts service health you have now specified the service policy.

CA Spectrum service manager provides a number of policies out-of-the-box, but you can create your own policies whenever an out-of-the-box policy does not match the resource behavior that you have identified.

You have determined the basic structure of the service, and can model it in CA Spectrum. You can consider this model to be a first-phase model. Chances are throughout this process you identified some areas where the monitoring capacity of the service is not adequate. As mentioned before service modeling is an iterative process, each phase expands, or refines the monitoring capacity of the service to make it more accurate.

The next section refining the service definition covers some of the common issues you will find when first defining a service, and provides some techniques for how to improve your service models.

## Refining the Service Definition

Regardless of the role that is defined for a service model, your goal can be to create a service such that the health of the service model accurately depicts the logical status of the service. You can eliminate scenarios where the service is logically down, but the service model indicates that it is up. Similarly, a service model cannot indicate that its health is impacted if logically the service is functioning normal. The following section introduces some techniques for refining your service models. These techniques are best applied in phases. Do not make too many enhancements at once. Define a strategy to improve your services, and implement that strategy. Break down the service revision process into multiple phases that can be managed and validated.

You can find various reasons for inaccuracy of service models monitoring capacity. This section covers a few of these reasons and discuss ways to improve your service models.

The following reasons are potential scenarios that can affect the accuracy of a service model:

- Missing resource models
- Resource models which are not discrete enough
- Resource faults that should not impact the service
- Service impacting events that do not impact the resource
- Patterns of resource faults which impact services

## Missing Resource Models

The most common issue when building service models is failure to identify all of a service resource. The principle reason for this is that most of the time we identify the resources that are providing the service directly to an end user, and not the capabilities that those resources rely on.

Continuing with the example of a web-based application, often when focusing on user faces applications, we start with the servers which host the application. This is a great starting point, but you may find that you have missed servers that support the application. Consider how the web application relies on a database to provide content. The following table displays the changes of our fault matrix:

Database	Server 1	Server 2	Service Health
Up	Up	Up	Up
Down	Up	Up	Down

Database	Server 1	Server 2	Service Health
Up	Up	Up	Up
Down	Down	Up	Down
Down	Up	Down	Down
Down	Down	Down	Down
Up	Down	Up	Slightly Degraded
Up	Up	Down	Slightly Degraded
Up	Down	Down	Down

The table is a bit more complicated, but you can see that with the addition of the Database, the previously identified rule set no longer works. You can see that regardless of the status of Server 1 and Server 2, if the Database is Down, the service is Down.

This type of pattern is common, and you have identified that the service relies on more than the web servers it requires extra resource which behaves differently than the web servers. To support this new set of resources, use a new model type named Resource Monitor.

The job of a resource monitor is to manage diverse sets of resources which do not follow a behavior pattern that can easily be captured in a single policy. A resource monitor is similar to a service in that it applies a policy to a set of resource values to determine its own health. It can be useful to think of resource monitors as a type of sub-service. A resource monitor by itself does not represent a logical service, but rather a critical aspect of that service.

In this example, you can see how resource monitors can be used. The following behavior is captured for the web servers and their relationship to the health of the service.

Server 1	Server 2	Service Health
Up	Up	Up
Down	Up	Slightly Degraded
Up	Down	Slightly Degraded
Down	Down	Down

If we specifically consider how the database impacts the service that a simple matrix can be created.

Database	Service Health
Up	Up
Down	Down

Capture each of these patterns into its own resource monitor.

The Database RM monitors the database resource with the following rule:

- When all resources are down the service is down

The Webserver RM monitors the web servers with this rule set:

- When any 1 resource is Down the service is Slightly Degraded
- When all resources are Down the service is Down

The service monitors the two resource monitors, and the service health reflects the status of the worst resource monitor. The following table show the fault matrix for the service:

Database RM	Webserver RM	Service Health
Up	Up	Up
Down	Up	Down
Down	Slightly Degraded	Down
Down	Down	Down
Up	Down	Down
Up	Slightly Degraded	Slightly Degraded

From this matrix you can see the following rule set for the service:

- When any 1 resource is Down the service is Down
- When any 1 resource is Slightly Degraded the service is Slightly Degraded

By expanding the service from monitoring the condition of two host models to instead monitoring the health of two resource monitors we have improved the accuracy of the service model. You can notice that the database resource is never referred to as a host model or a database server. Because the database is likely to be a service itself.

This scenario of missing resources is common that you can consider it when creating service models. Even if the database server was not identified as a service resource in the initial phase the service could still have been created to monitor a single resource monitor for the web servers. In later phases, as additional resources are identified, it is easy to add new resource monitors to the service itself.

## Resource Models Which Are Not Discrete Enough

Missing service resources is certainly one common cause for inaccurate service model. Another cause for inaccuracy in services model is resource models which are not discrete enough. Usually CA Spectrum monitors devices at a fairly high level, and reports their Condition that is based on fairly basic criteria. For example, a host model which responds to SNMP traffic and has normal CPU and memory utilization can be considered Up or Normal.

Going back to the web application service example, consider if the simple host monitoring is adequate to determine if the web application is working. Imagine if on webserver 1, a critical file system has become full, and on webserver to the actual webserver process was not running. In this case, the web application service is not available to users, but the given basic host monitoring the service model indicates a health of Up.

In addition to the host being contactable, various other aspects can be monitored to determine if the web application is running. To determine the application status, you can use the following common components:

- Monitoring critical processes
- Monitoring critical file systems
- monitoring application connectivity and response time

Looking at each web server, you can define a fault matrix using the following example:

Host	Process	File System	App Connection	Service Health
Up	Up	Up	Up	Up
Down	Down	Down	Down	Down
Up	Down	Up	Down	Down
Up	Up	Down	Up	Down
Up	Up	Up	Down	Down

In the preceding abbreviated matrix, you can notice that for the web server to be considered up, consider more than simply a host model. A webserver process, a critical file system, and the web application can also be responsive to connections and requests.

You can notice that rather than simply a host model each webserver is actually a service in and of itself. From the preceding example, you can envision a service with three resource monitors.

The Host RM, simply monitors the Condition of the host model. The Proc and FS RM monitors the Condition of a process model, and a file system model. The App Conn RM monitors the status of a series of response time tests which send requests to the server.

At first, given that the Host RM and the Proc and FS RM are both monitoring the Condition attribute, you can combine them into a single resource monitor. You can separate the resource models into two resource monitors as they represent different classes of models. In the next section, you can notice, isolating the host model within its own resource monitor that gives you the ability to exclude host-related alarms (that do not impact the service).

The Host RM and Proc and FS RM models have simple fault matrix tables.

For the Host RM model:

Host	Service Health
Up	Up
Down	Down

For the Proc and FS RM Model:

Process	File System	Service Health
Up	Up	Up
Down	Up	Down
Up	Down	Down
Down	Down	Down

The App Conn RM is based on response time monitoring. It is at your discretion to use the multi-level threshold capability of CA Spectrum service performance manager to create more health values. For simplicity, look at a Timeout, Critical Threshold, and Major Threshold configuration. The following matrix looks like a condensed set of response time test values:

Response Time	Service Health
Normal	Up
Timeout	Down
Critical Threshold	Down
Major Threshold	Degraded

Consider the following fault matrix for a single webserver:

Host RM	Proc and FS RM	App Conn RM	Service Health
Up	Up	Up	Up
Down	Down	Down	Down
Up	Down	Down	Down
Up	Up	Down	Down
Up	Up	Degraded	Degraded

You can determine that all resource monitors must be Up for the web server to be considered Up. Remember, this is the fault behavior for a single web server.

In review we earlier expanded the web application service to include two resource monitors: Database RM, and Webserver RM. At the time, the webserver RM was monitoring the Condition of two host models. From the preceding matrix, you can identify that rather than two host models, the webserver can be represented as two service models each with three resource monitors. The previous configuration still applies, but now the webserver RM monitors the service health attribute of two services models, instead of the Condition of two hosts. You can see that despite vastly improving the accuracy of the web application service the structure of the service remains intact. The fault matrix that is determined early is still accurate:

Database RM	Webserver RM	Service Health
Up	Up	Up
Down	Up	Down



Database RM	Webserver RM	Service Health
Down	Slightly Degraded	Down
Down	Down	Down
Up	Down	Down
Up	Slightly Degraded	Slightly Degraded

What has been improved is the definition of Up for the webserver RM.

## Resource Faults that Shouldn't Impact the Service

Missing resources and resources that are not discrete enough can often lead to a service indicating a health of Up when it should not. Sometimes a service model may indicate a health of Down or Degraded when logically it is not. This can best be described as a non-service impacting resource fault.

Two common situations are available that produce this type of problem, both of which are different.

The first situation is when a resource models status is influenced by associated child models. This situation can happen, if you select a model that is logically a service resource but has a status that can be affected by other models (that are not service resources). For example, when you specify a network device as a resource for a particular service, rather than the interfaces that actually support the service. Another example, when dealing with host models or servers to specify that the host itself alarms for a failed process or file system instead of the monitored process or file system model. If the process or file system is not logically a part of the particular service, the service may indicate an affected health when it should not. Such situations are easy to resolve by selecting the most appropriate model as the service resource, for example, a specific interface, or monitored process.

The second situation occurs exclusively when a service monitors the Condition of its resource models. The Condition of a resource model holds the value of the most severe alarm on the model. CA Spectrum generates thousands of different types of alarms. Some of these alarms are indicative of service impacting faults, but many are not. By default CA Spectrum monitors devices for various reasons, but principally to insure that are functioning correctly from a network infrastructure perspective. You can see various alarms which faults having to do with network management, but these alarms may have no logical impact on capability that is being monitored by the service. Similarly, a given model can be a resource of multiple services, such that a particular alarm can be significant for some services and not others. This occurs frequently when the purpose of the service is to monitor compliance that is based on some specific responsibility.

Service Manager alarm type exemption functionality is designed to support resource alarms which should not impact service health. When you discover that certain resource faults should not impact a service, you can specify an alarm type exemption configuration. This configuration can be specified for an individual service or resource monitor, or if the behavior is common it can be specified in a service policy.

By applying alarm type exemptions, you can again improve the accuracy of your service models by eliminating false service outages.

## Service Impacting Events That Don't Impact Resources

Sometimes, you can encounter a situation where a resource model experiences an event that does not produce an alarm on the resource. However, it logically affects the capability being monitored by the service. When deployed with a primary focus on network infrastructure management, often many events that are produced by or passed to CA Spectrum which do not produce alarms. Because the events are too common or insignificant. That said there are circumstances where these events are ignored and can be considered for a specific service. In some situations, it can be appropriate for you to map the event to a new alarm, which affects the Condition of a resource model. As the alarm configuration takes affect for all models of a given type, it may not be a good practice if the event is significant only for a few models.

When the situation arises that an event is service impacting, but does not produce an alarm, you can consider creating a Correlation Domain to represent the service resource. Much like a service determines its own health by monitoring attributes of its resources, a correlation domain can determine its own condition by monitoring events occurring on its resources.

To support non-alarm producing events, create a Correlation Condition for the event. You can perform this task by specifying the event code for the logical set, and then the corresponding event code for the logical clear.

Revisit the Database service that discussed early in this guide, for example, the Database service is comprised of two host models. At some time during the day, each database server can initiate an automatic data archival process. Consider that CA Spectrum was integrated with a database management tool which can produce an event on the CA Spectrum host model when the archival process begins and another event when the archival process is complete. Since this behavior is normal for the server, there is no alarm. However, during the archival process the server may not respond as quickly to requests.

In general, the data archival process does not impact the database service as a whole. However, if both database servers were archiving data simultaneously, the service becomes Slightly Degraded. If one of the database servers is down, and the other is archiving data, the service becomes Degraded.

You can create a Correlation Condition and a pair of Correlation Rules to capture this behavior.

The Correlation Condition consists of a set event code and clear event code which correspond to the start and completion of the archival process.

Two new Correlation rules can be created which specify an Implied Cause of either Service Impact Slightly Degraded or Service Impact Degraded. Service Manager adds these out-of-the-box Conditions with a Service Impact Down Condition. Consider the following correlation rules:

- DB Archiving Exists AND Device Contact Lost Exists Implied Cause Service Impact Degraded
- DB Archiving Count = 2 Implied Cause Service Impact Slightly Degraded

Both rules specify the Correlation Domain as the root cause target. The new rules can be combined into a new Correlation Policy that is named Data Archival Policy. You can create a Correlation Domain using the Data Archival Policy, and can specify both database servers as resources.

Finally to extend the Database service, a new resource monitor, Data Archival RM can be created to monitor the Condition of the correlation domain.

By using this approach, you are able to show the service impact of non-alarm producing events.

## Patterns of Resource Faults Which Impact Services

You can encounter resource monitoring scenarios where the relationship of particular service resources is too complex to capture simply by monitoring the Condition attribute of the models. More specifically, the Condition of the model can have a greater or lesser significance depending on some additional resource monitoring criteria.

Consider the following scenario. An account management team working in a remote office uses a local database server to access customer information. If the local system is down, the account management team can access the information that they require by connecting to a server at the company head quarters. The service representing the customer account system behaves in the following manner:

- If the local server is down, and the head quarters server is up, the customer account service can be Slightly Degraded
- If the local server is up and the head quarters server is down, the customer account service can be Up.
- If the local server is down and the head quarters server is down, the customer account service can be Down
- If the local server is down and the head quarters server is in maintenance, the customer account service can be Down
- If the local server is in maintenance and the head quarters server is in maintenance, the customer account service can be Down

As you can see, the service impacting scenarios are fairly complex. Even though there are two servers providing the service, the servers are not treated equally.

For handling a resource monitoring scenario, such as Correlation Domain can be used that manages the complexities of monitoring the individual resources.

A service utilizing two resource monitors can be used to implement this example. First, you can isolate the local server into its own resource monitor, which uses a simple fault matrix. The Local Server RM uses a policy to support this pattern.

Local Server	Service Health
Up	Up
Down	Slightly Degraded

The second resource monitor that is Local and Remote Domain RM also has a simple fault matrix:

Local & Remote Domain	Service Health
Up	Up
Down	Down

The Local and Remote Domain RM, monitors a Correlation Domain which contains the host model for the local server and the host model for the head quarters server.

Verify the following status table of the Condition of the Correlation Domain:

Local Server	Remote Server	Domain Condition
Normal	Normal	Normal
Contact Lost	Normal	Normal
Normal	Contact Lost	Normal
Contact Lost	Contact Lost	Critical
Contact Lost	Maintenance	Critical
Maintenance	Maintenance	Critical
Maintenance	Contact Lost	Critical

The domain uses a policy with the following rules, all utilizing the domain for the root cause target.

- Device Contact Lost Count = 2 Implied Cause Service Impact Down
- DeviceInMaintenance Count = 2 Implied Cause Service Impact Down
- Device Contact Lost Exists AND DeviceInMaintenance Exists AND Device Contact Lost Model Does Not Equal DeviceInMaintenance Model Implied Cause Service Impact Down

The Customer Account Service monitors the service health of the two resource monitors in the following way:

Local Server RM	Local and Remote Domain	Server Health
Up	Up	Up
Slightly Degraded	Up	Slightly Degraded
Slightly Degraded	Down	Down
Up	Down	Down

The Customer Account Service bases its health on the worst status of the two resource monitors which is the common high sensitivity pattern.

Using this approach, you can create accurate resource monitoring capacity for your services even when the scenarios have complex requirements.

## Service Attributes and Relationships

When creating a service, specify a number of attributes and associate the service to its resources and other service management models. This section explains the attributes that can be configured and the potential relationships that you can create for each service model.

### Name and Description

The service name and description identify the service model. You can define multiple services with the same name, provided they have unique descriptions.

You do not have rules for defining service names. However, you are recommended to use some naming scheme or convention that lets you identify a service model, if you encounter it outside of the service management hierarchy. Some naming schemes include multiple parts that let you categorize the service geographically or organizationally. Other naming schemes associate the service to a particular customer or function.

It is a useful practice to state the service role in the service description. The service role is determined in the planning phase, and states the capability and purpose of the service model.

### Criticality

Service criticality is an enumerated value ranging from a Low value of 10 to a High value of 30. All or a portion of a services criticality can be added to the Impact value of any resource alarms effecting the service.

If a service is down, the entire value is factored into the calculation. If a service is degraded as a result of the resource alarm, one half of the service criticality value is factored into the impact calculation for the alarm. If a service is slightly degraded, one fifth of the value is factored into the calculation.

Verify the following Criticality values:

- Low (default) = 10
- Medium Low = 15
- Medium = 20
- Medium High = 25
- High = 30

For example, a Degraded High criticality service has an impact of (50percent) \* 30 = 15, and a Down Low criticality service has an impact of (100percent) \* 10 = 10. The alarm that caused the Degraded High criticality service has a comparatively greater impact than the alarm the caused the Down Low criticality service. If alarms are sorted and prioritized by impact value, the alarms impacting the most critical services are given the highest priority.

### Landscape

The landscape field specifies the landscape where the service model resides. The landscape option appears only when multiple SpectroSERVERs are deployed in a distributed environment.

To optimize performance, the service can be created on the landscape where all or most of its resources reside. If the service has resources spanning multiple landscapes review the section of this guide entitled Service Models in a DSS environment.

### Security String

The security string secures access to the service model in CA Spectrum. For more information, see the *CA Spectrum Administrator Guide*.

**Note:** Security in the service dashboard differs from the OneClick Console. If a user does not have access to a service model, all icons and list entries for that service are absent from the service dashboard. This dashboard differs from the OneClick console where icons are exposed, but no model data is available.

### Maintenance Mode

Service models support maintenance mode as do many other models in CA Spectrum. When a service is "in maintenance", it is not actively monitoring any of its resources.

You can create a service in maintenance to avoid the generation of any service outages while you are still building your service hierarchy and identifying resources. In this capacity maintenance, model is used to show that the service is under construction.

Service models also support scheduled maintenance. Schedule maintenance defines preconfigured periods of time where the service stops actively monitoring its resources. Commonly service level agreements specify periods of time which are reserved for service maintenance.

### Generate Service Alarms

Each service can be configured to generate alarms which correspond to changes in service health.

Disabling the generation of alarms for the service means that an alarm cannot be generated, the service health is still modified based on policy evaluation. All icons within the OneClick Console and Service Dashboard shows the appropriate color for service health regardless of whether an alarm is generated. Regardless of the generate service alarms setting, all or a portion of the services criticality are added to any resource alarms impacting the health of the service. The service is shown in the service impact table of such resource alarms, even if no alarm is generated for the service model itself. Any guarantee models associated to the service tracks outage time regardless of whether an alarm is generated for the outage.

There are several reasons to disable the generation of service alarms. First is to reduce the number of alarms that are produced in CA Spectrum. If you are sure that all resource alarms indicate the service impact, then the service alarm can be unnecessary.



Another reason to disable service alarms is when the alarm is redundant. Often multiple services are created which monitor many of the same resources, but with a different role. For example, you can create a service model which focuses on the real-time status of a service and can have it generate alarms. Other services models monitor specific aspects or resources of the service model for SLA purposes and can be configured such that they do not produce an alarm.

### **Containers**

The containers setting specifies how a service monitors its resources, if the specified resource is a type of container model. This setting applies only to those resources which are containers and is not used for non-container resources.

You can add different types of containers to a service. Depending on the type of container, you can configure the service to monitor container model itself or the contents of the container. When set to Monitor Contents (default), the service applies the policy to the models within the container. When using Monitor Container, the service applies the policy to the container itself.

When Monitor Contents is specified the service monitors the containment relations of the container model, and updates its resources as models are added or removed from the container.

Consider, for example, the effect of physically removing a router and replacing it with a new router during a network upgrade. If the original router model was placed in a container model that is monitored by Service Manager, it automatically removes from the service. If you place the new router in the same container, it automatically monitors as part of the service.

Many environments tend to be dynamic (the service resources can periodically change). Consider the addition of new infrastructure components that increase capacity and mitigate the risk of failure. As these resources are added services, you can take them into account. Structured containers and services can adapt easily to these types of changes.

You can view a list of containers in a service under Containers Providing Resources in the OneClick Console or Service Dashboard Information view. Verify the following image:

File View Tools Help

Service 1 of type SM\_Service

Information Host Configuration Root Cause Interfaces Performance Neighbors Alarms Events Attributes

Service 1  
Service

Service 1  
[set](#)

+ General Information

+ Outage History

+ Resources

- Containers Providing Resources 

Filter: 
Displaying 0 of 0

Condition ^	Name ^	Type	Child Count

+ Customers

The Containers Providing Resources subview indicates the container condition, name, type, and child count (the number of resources in the container).

The Containers setting applies to all resources which are derived from the Container model type. If you want different behavior for different containers within the same service, consider the use of multiple resource monitors, which each has their own container monitoring setting.

### **Service Policy**

During the planning phase, you identified the resource attribute to monitor and the rule set by which to determine service health. You can select an existing policy that matches this behavior or can create a new one.

The policy should reflect the behavior of the resources that the service monitors. You can select the policy first by which resource attribute can be monitored, and next by which policy best reflects the behavior of the resources collectively. For example, if the service monitors a pair of redundant servers, a redundancy type policy would be appropriate. If you find no single policy accurately reflects the behavior of the resource, you can create multiple resource monitors to organize resources that are based on their specific monitoring requirements.

Service Manager provides a set of standard policies that represent common service monitoring requirements. It also lets you create custom policies to meet your particular requirements.

If the service uses resource monitors, or have other services as its resources only Service Health based policies can be used.

## **Create a Service**

Have an understanding of service and the resources that provide the service, before creating a service. Consider how the resources can be monitored and the relative impact of resource outages on the availability of the service. Identifying the resources and how they impact the service makes policy and resource selection easy.

Services that you create appear under the Service tab in Service Editor, in Service Dashboard, and in OneClick Console under Service Management in the Navigation panel.

### **Follow these steps:**

1. Open the Service Editor.
2. Click Tools, Utilities, Service Editor from the main menu.

3. Click the Services tab.

At the bottom of Services table you can see the following options/tabs:

- List - Creates a service as a child of the top-level services model
- Hierarchy - Creates a service as a child of the service which is selected when the Create button is clicked

If you are going to create a service as a child or resource of another service the parent service must use a Service Health based policy.

4. Click Create.

The Create Service dialog appears. The Create Service dialog lets you specify service properties, the IT resources that support the service, and the service policy or resource monitors that define which resource attributes are monitored.

5. Specify the following service properties, some are required for service creation other are optional. Most required fields have default values, but always consider if these values are appropriate.

### **Name (Required)**

Multiple services can have the same name provided they have unique descriptions. Consider using a naming scheme or convention that allows for quick identification of the service model.

### **Description (Optional)**

Describes the service. You can enter unique descriptions for services that have the same name to facilitate finding each service using a list filtering utility. The service description appears within Service Availability and Service Health Reports.

### **Criticality (Required)**

Specifies the criticality value that is factored into the impact calculation for a service resource model in an alarm state (the root cause alarm).

### **Landscape (Required)**

Specifies the landscape where the service model can be created. The landscape field appears only when you are working in the DSS environment.

### **Security String (Optional)**

Specifies the security string for the service model.

### **In Maintenance (Optional)**

Selecting this option puts the service model into maintenance mode.

### **Generate Service Alarms (Required)**

Specifies whether CA Spectrum generates alarms for the service model which correspond to changes in service health.

**Containers (Required)**

Specify how the service should monitor resources which are containers.

6. Specify the service policy and click the Set button to select or create the service policy.

For services that define resource monitors or monitor of have other services as their resource only Service Health based policies can be used.

**Note:** When you specify Service Policy, the Service Editor automatically selects the most appropriate table to display the resources of the service.

**Note:** The service policy dictates which attribute of the resources can be monitored. Make sure the service resources that you specify support the attribute that you select. If a resource is specified which does not support the attribute, SPECTRUM generates a yellow (minor) alarm for the Service Management model.

7. If the service uses resource monitors, click the gear icon to create each resource monitor.

For each resource monitor, specify a name and policy. You can also specify the Container behavior, and an alarm type exemption.

8. Configure alarm type exemptions.

This feature is only available for services which are using a Condition-based Policy.

9. Select containers and resources for a service by clicking the binoculars icon to launch the resource locator; search for resources and add them to the service.

If you have created resource monitors first select the resource monitor, and then launch the resource locator.

When specifying resources which are container mode, the Containers configuration that is specified in step 5 applies. For more information about working with searches, see the *CA Spectrum Administrator Guide*.

If you are creating the service on the main location server, select resources from any landscape. If the service is created on a landscape which is not the MLS, you are restricted to select resources only from the local landscape, and the MLS. Notify your organization CA Spectrum administrator if the search does not find the resources that you expect to find.

10. Select the resources (models) you want to include from the returned search list and click Add Selected to Monitored Resources. Close the Locate Resources dialog when you have all resource.
11. Click the Create button to create the service model.

The service now appears in the table view of the service editor.

**More information:**

[Service Health Policies](#) (see page 189)

## Resource Monitors

### Resources in Maintenance Mode

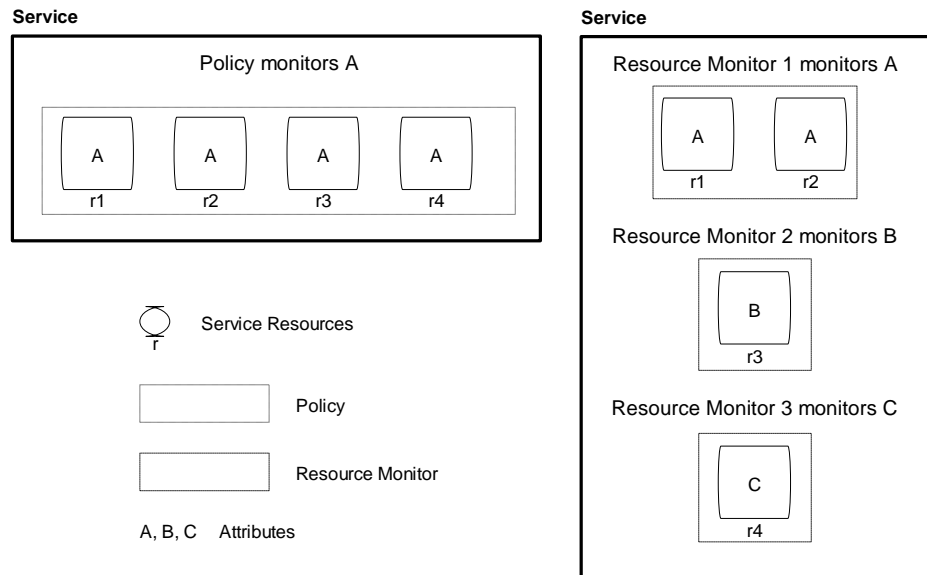
When a service resource is put into maintenance mode, Service Manager stops monitoring the resource during the maintenance mode period. For certain types of resources (port models, monitored process models, and monitored file system models, for example), Service Manager respects the maintenance status of the parent device model.

When the parent model for the following resource types is put into maintenance mode, the service stops monitoring the resource:

- interface models
- monitored process models
- monitored disk (file system) models

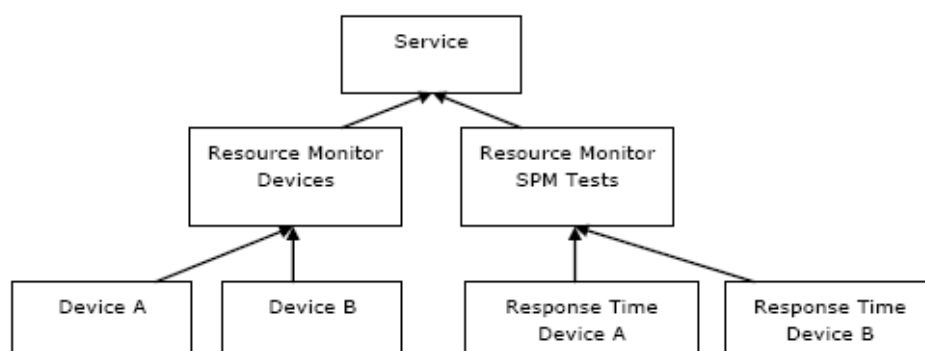
A resource monitor is a CA Spectrum model. A policy, which is not a model, specifies a *single* watched attribute common to *all* resources monitored by a service, a resource monitor monitors an *attribute* common to *one or more* service resources to determine its own service health status, the same way a service determines its health status. Using resource monitors lets you implement a finer mode of monitoring service health than is provided by a single policy.

The following diagram illustrates the difference between the two by showing how each could be implemented for the same service:



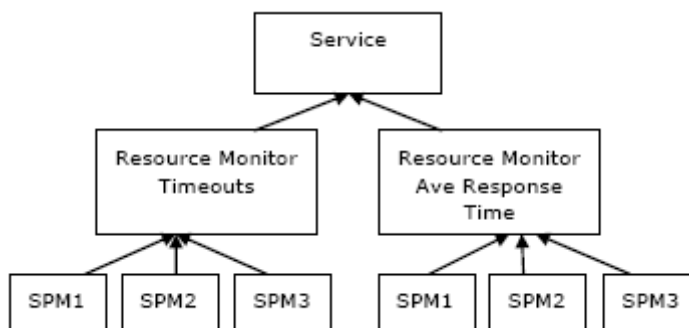
A resource monitor is a resource of a service, and you can apply multiple resource monitors to monitor a service. A service monitors the service health attribute of each resource monitor to determine its own health.

A simple scenario of when resource monitors can be used is when the resources of a service are of mixed types. For example, when monitoring a pair of device models, and a pair of response time test models. Different policies are used to monitor device models, and SPM test models, resource monitors could be created to organize the device models, and the SPM test models in the following way:



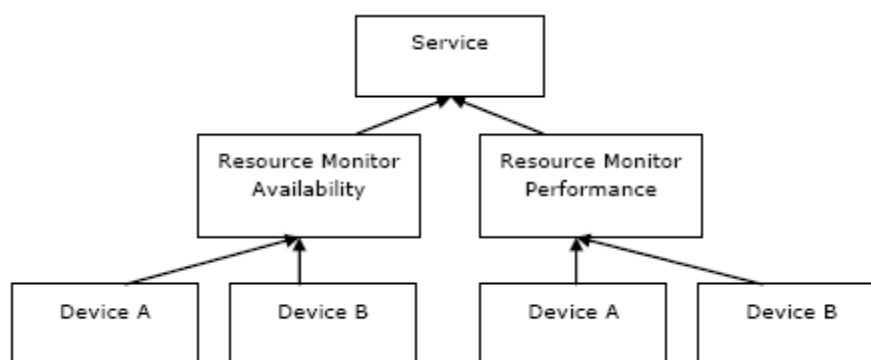
The Service model would use a policy such as Service Health High Sensitivity, the Resource Monitor for Devices uses the policy Condition Redundancy, and the Resource Monitor for Response Time tests uses Response Time Redundancy.

In other cases, you can monitor multiple attributes of the same resources. In other words apply multiple policies to the same resources. For example, you may be interested in monitoring a set of SPM Tests with a response time policy such that the service is impacted when too many response time tests are timing out. You also want to monitor the average response time of the same response time tests such that the service is impacted when the average response time exceeds some specific threshold. Again resource monitors can be used to create this type of service.



In this scenario, the service uses the Service Health High Sensitivity policy. The Resource Monitor for timeouts could use a custom policy focusing specifically on the Latest Error Status value of Timeout for the SPM Tests. The Resource Monitor for average response time uses another custom policy which compares the average Latest Result for the SPM tests to a particular set of threshold values.

In yet another scenario, multiple resource monitors can monitor the Condition of common resources, but can use different policies to organize resource outages by fault type. For example, resource monitors could be used to categorize availability impacting faults and performance impacting faults:



In this configuration, the Service uses a service health policy, the availability resource monitor uses a customer policy with an Alarm Type exemption configuration designed to isolate resource faults that are designated as availability impacting. The performance resource monitor would in turn use of custom policy which isolates resource faults that are designated as performance impacting.

## Create a Resource Monitor

You can monitor multiple attributes on a single resource with different resource monitors.

### Follow these steps:

1. [Open the Service Editor](#) (see page 18).
2. Click the Services tab and click Create.

The Create Service dialog appears.

3. Click .

The Create Resource Monitor dialog appears.



4. Perform the following steps:
  - a. Enter a name for the resource monitor and, optionally, a security string.
  - b. Select whether the resource monitor monitors the resources in a container or the container itself.
  - c. Select a policy.
  - d. (Optional) [Specify which alarms you want to affect or not affect the health of the resource monitor](#) (see page 57).
  - e. Click OK.

The resource monitor appears in the Containers in Resources to Monitor list in the Create Service dialog.

5. Add the resources with the attribute you want to monitor (the attribute that is specified by the resource monitor policy) with the resource monitor:
  - a. Select the resource monitor.
  - b. Click the Locate resources and containers icon.
  - c. Select the containers and resources for the resource monitor.
  - d. Click Add Selected to the Monitored Resources.

The resources appear under the resource monitor icon in the Containers and Resources to the Monitor panel.

## Specify the Alarm Types That Affect or Do Not Affect Service Health

You can specify resource alarm types to affect or not affect service health when you create a service or resource monitor with a policy that specifies the Condition attribute. You can perform this task from the Exemptions Panel by selecting the appropriate alarm impact options.

**Note:** This functionality replaces the manual setting of the Exempt\_Cause\_List attribute which was documented for previous releases.

Consider the following information before you specify resource alarm types:

- Individual services or resource monitors can specify a list of alarm types to be excluded from or included in the service health calculation.
- Service alarm type exemptions can be used to establish a specific behavior for individual services. For example, only this service is affected by this alarm type, or this particular service should not be affected by this alarm type.
- Service alarm type exemptions take precedence over any configuration that is defined at the policy.

**Note:** When you specify alarm type exemptions for a service, Service Manager ignores any exemption specification that is defined for the policy that is used by the service.

**Follow these steps:**

1. Perform one of the following tasks:
  - [Create a service](#) (see page 51).
  - [Create a resource monitor](#) (see page 56).
2. Click the Exemptions tab and select one of the following alarm impact options from the Service Health Impacted Only When Resource Outages \* drop-down list:

**Caused By**

Only these selected alarm types impact the service health.

**Not Caused By**

Exclude these alarms from impacting the service.

**Disabled (default)**

Do not use service level alarm type exemptions. If the policy defines exemptions, you can use it.

3. Move the available alarm types that you want to affect or not affect a service to the Selected Alarm Types box, or specify a range of alarm cause codes.

The alarm types are specified.

**More information:**

[Create a Policy](#) (see page 77)

[Example: Define an Alarm Exemption List for a Service or Resource Monitor](#) (see page 133)

## Alarm Filters

In CA Spectrum the health of a service can be impacted based on the alarms that are generated on the resources on which the service is monitored. If a filter is applied on any specific alarm, that particular alarm participates in affecting the health of the service. The alarm is also considered while calculating the health of a service.

The following image shows the Alarms Filter tab and the available options:

**Create Service - CA Spectrum OneClick**

Name \*

Criticality \* Low

Security String

In Maintenance No

Generate Service Alarms Yes

Containers \* Monitor Contents

Description

Service Policy \* Condition High Sensitivity Select...

Resources Alarm Filters

Service Health is impacted when resource outages \* caused by:

Select Individual Alarm Types

**Available Alarm Types**

- (0x4b60052)
- (0x1169b18)
- % POOL BUSY HEALTH INDEX (0x11029)
- % POOL BUSY TREND (0x11066)
- ( BOOT ) PARTITION DISK USAGE HIGH (0x5c30033)
- ( CBCONFIG ) PARTITION DISK USAGE HIGH (0x5c3003E)
- ( ROOT ) PARTITION DISK USAGE HIGH (0x5c30031)

Filter:  Displaying 21,838 of 21,838

Refresh

**Selected Alarm Types**

Filter:  Displaying 0 of 0

Specify Optional Alarm Cause Code Ranges

Alarm Cause Code Ranges:

If the alarm filters tab is disabled, the alarm functionality does not work, irrespective of alarm types that are selected to impact the health of the service. By default, the alarm filters tab is disabled.

### **Service Health is impacted when resource outages\***

#### **caused by**

Impacts the health of the service if the generated alarm on Service Monitored Resources (SMR) is of the selected alarm type.

#### **not caused by**

Impacts the health of the service if the generated alarm on SMR is not of the selected alarm type.

### **Alarm Cause Code Ranges**

Impacts the health of the service if the generated alarm on SMR falls in the specified alarm cause code range.

### **Specify Alarm Title Keywords**

Impacts the health of the service if the generated alarm title on SMR matches with the specified alarm title. For example, CHASIS.

**Note:** You can define the same functionality (Alarm Title Keyword, Alarm Types) on Alarm Filter at the service policy level.

## Add a Resource to a Service

You can create a service and can add resources to the service, or can add resources to an existing service or resource monitor using the OneClick Console as an alternative to using the Service Editor.

### **Follow these steps:**

1. Select the resource (model) you want to add to an existing service or a service you plan to create.
2. Right-click the model and click Add To, Service.

The Add to Service/Resource Monitor dialog appears. This dialog lists available services.

**Note:** To display resource monitors for a service, select the service.

3. Create a service to which you can add the resource or can add the resource to an existing service or resource monitor:

- To create a service, click Create.

The Create Service dialog appears. It lists the selected resource as a resource of the service. Specify service properties as described in [Create a Service](#) (see page 51).

- To add the selected resource to an existing service or a resource monitor, select the service or resource monitor to which you want to add the resource and click OK.

The resource is added to the service or resource monitor.

## Delete a Resource from a Service

You can remove a resource from a service using the OneClick Console as an alternative to using the Service Editor.

### Follow these steps:

1. Select the resource (model) that you want to remove from an existing service.
2. Right-click the model and click Remove From, Service.
3. Click OK.

The resource is removed from the service.

## Edit a Service

You can change all service properties anytime after you create a service.

Consider the following information before you edit a service:

- If you change a service name, all historical data in reports for the service that is generated with CA Spectrum Report Manager is listed under the new name.
- If you delete a service that is monitored by an SLA, the SLA guarantee goes to the Initial (Blue) state.
- If a service has a resource monitor monitored by an SLA and you delete the resource monitor, the SLA guarantee watching it goes to the initial (Blue) state.

**Important!** Editing services can be performed by authorized personnel who are aware of the potential ramifications of these actions, particularly as they relate to services monitored by SLAs.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Services tab, select the service that you want to edit from the list, and then click Edit.

The Edit Service dialog appears.

Edit the settings, as described in [Create a Service](#) (see page 51), and click OK.

The service is edited.

## Delete a Service

Before you delete a service, verify that it is not associated with an SLA you intend to implement that is scheduled to be activated.

**Important!** Deleting services can be performed by authorized personnel who are aware of the potential ramifications of these actions, particularly as they relate to services monitored by SLAs.

**Follow these steps:**

1. [Open the Service Editor dialog](#) (see page 18).
2. Click the Services tab, select the service that you want to remove from CA Spectrum, and then click Delete.
3. Respond to the confirmation message that appears to complete the deletion.

The service is deleted.

## Cut a Service

CA Spectrum Service Manager lets you cut a resource monitor as part of editing a service.

Create a service and add a resource monitor. The policy and resources do not matter. Save and edit the service.

**Note:** With the resource monitor selected, the cut icon is disabled.

**Follow these steps:**

1. [Open the Service Editor dialog](#) (see page 18).
2. Click the Services tab, select the service that you want to cut from the resource monitor, and then click Cut.

The service is cut.

## Service Maintenance Schedule Management

When you schedule maintenance for a service, CA Spectrum puts the service in maintenance mode (brown state) for the duration that is specified by the schedule. A service in maintenance mode is a planned outage. If the service is monitored by an SLA, the frequency and duration of the scheduled planned outages are typically defined by the SLA contract between a service provider and a service customer. Planned service outages are not accumulated as down or degraded time by SLA guarantees.

You can manage service maintenance schedules in the following ways:

- Create and save multiple one-time and recurring schedules.
- Add schedules to the list of schedules to be implemented on an as-desired basis.
- Remove schedules from the list of schedules to be implemented.

**More information:**

[Example: Define a Service Maintenance Schedule](#) (see page 133)

### Create a Maintenance Schedule

When you create a schedule, Service Manager saves it to CA Spectrum. The schedule is added by default to the list of schedules that can be implemented for the service. If you do not want the schedule to be implemented, you can remove it from the list and can add it to the list at another time. You have two ways to create or specify a maintenance schedule for a service. You can specify scheduled maintenance from the Component Detail view of a service model much like other CA Spectrum models, or you can configure schedule maintenance from the service editor.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Services tab, select the service for which you want to manage maintenance schedules, and then click the Scheduled Maintenance tab.
3. Click Create.  
The Create Schedule dialog appears.
4. Configure the new schedule and click OK.  
Service Manager adds the schedule to the Current Schedules list.

**More information:**

[Remove a Maintenance Schedule from the Current Schedules List](#) (see page 64)

[Add a Maintenance Schedule to the Current Schedules List](#) (see page 64)

## Add a Maintenance Schedule to the Current Schedules List

If you want to implement a maintenance schedule for a service, include it in the Current Schedules list for the service.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Services tab, select the service for which you want to manage maintenance schedules, and then click the Scheduled Maintenance tab.
3. Select the schedule to add from the Available Schedules list and move it to the Current Schedules list.

The maintenance schedule is added to the Current Schedules list.

## Remove a Maintenance Schedule from the Current Schedules List

If you do not want to implement a maintenance schedule, you can remove it from the Current Schedules list.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Services tab, select the service for which you want to manage maintenance schedules, and then click the Scheduled Maintenance tab.
3. Select the schedule that you want to remove from the Current Schedules list, and then move it to the Available Schedules list.

The maintenance schedule is removed from the Current Schedules list.



## Associate an Owner with a Service

Service Manager lets you designate one or more users as owners of a service. A service owner serves as the contact person for the service. Service owner information and also be available from Service Availability and Service Health reports.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Services tab, select the service to associate with an owner, click the Owners tab, and then click Select Service Owners.

The Select Owners dialog appears.

3. Move the owner to associate with the service from the Available Owners list to the Service Owners list and click OK.

The owner is associated with a service.

## Associate a Customer with a Service

Service Manager lets you associate one or more customer models with a service to help you track and manage services and customers. You can generate service reports with CA Spectrum Report Manager based on service customer associations. When a service is associated to one or more customer models, the Criticality of each customer can be factored into the alarm impact for any service impacting outage. This assures that resource outages which impact highly critical customers can have a greater impact value.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Services tab, select the service that you want to associate with a customer, click the Customers tab, and then click Select Service Customers.
3. The Select Customers dialog appears.
4. Move the customer that you want to associate with the service from the Available Customers list to the Customers that use this Service list and click OK.

The customer is associated with a service.

## Service Models in a DSS Environment

When creating service models in a distributed SpectroSERVER (DSS) environment there are some important factors to consider. Service models can be associated to resource models from other landscapes. The supported behaviors are as follows:

- A service model that is created on the MLS can monitor resources from the MLS or any second tier landscape.
- A service model that is created on a second tier landscape can monitor resources models on its own landscape or the MLS, but not other second tier landscapes.

The following behaviors apply to services which use global collections to define their resources:

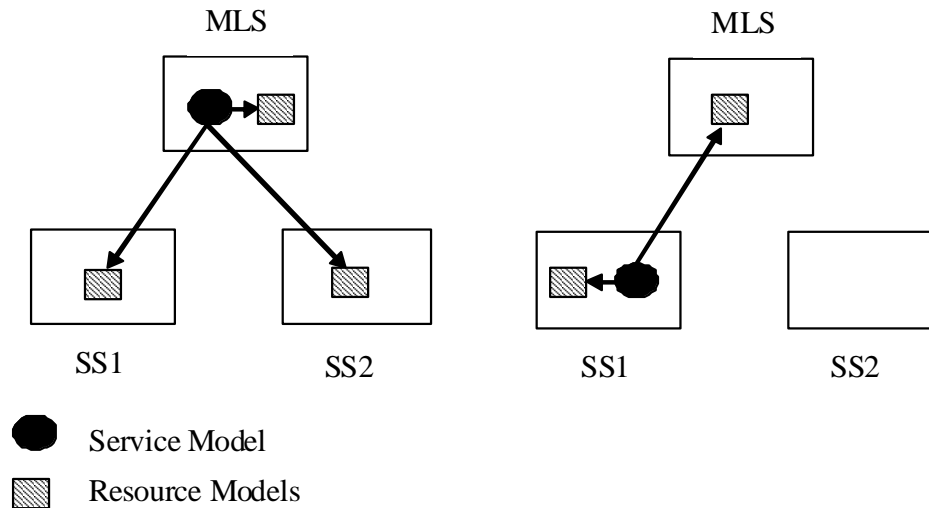
- A service model that is created on the MLS using a global collection can monitor all resources that are contained by the global collection.
- A service model that is created on a second tier landscape using a global collection can monitor only those resources in the collection which reside on its own landscape or the MLS.

In CA Spectrum r9.2, the performance heavy cross landscape watches is replaced with asynchronous action notifications, relieving some of the burden from the SpectroSERVERs. In addition, a relay mechanism allowing the MLS to forward second tier notifications from one landscape to another allows second tier services to monitor resources from other second tier landscapes.

This lets you construct services with less concern over performance impact. In addition, it lets you create services on landscapes where the service should logically reside without concern over what landscape required remote resources reside upon. This does not eliminate the need for an efficiently designed service hierarchy, but it provides flexibility which makes it easier to design the service hierarchy. All resources from all landscapes are visible, regardless of where the service mode is being created.

## Example: Supported Service to Remote Resource Configurations

The following image depicts the supported service to remote resource configurations:



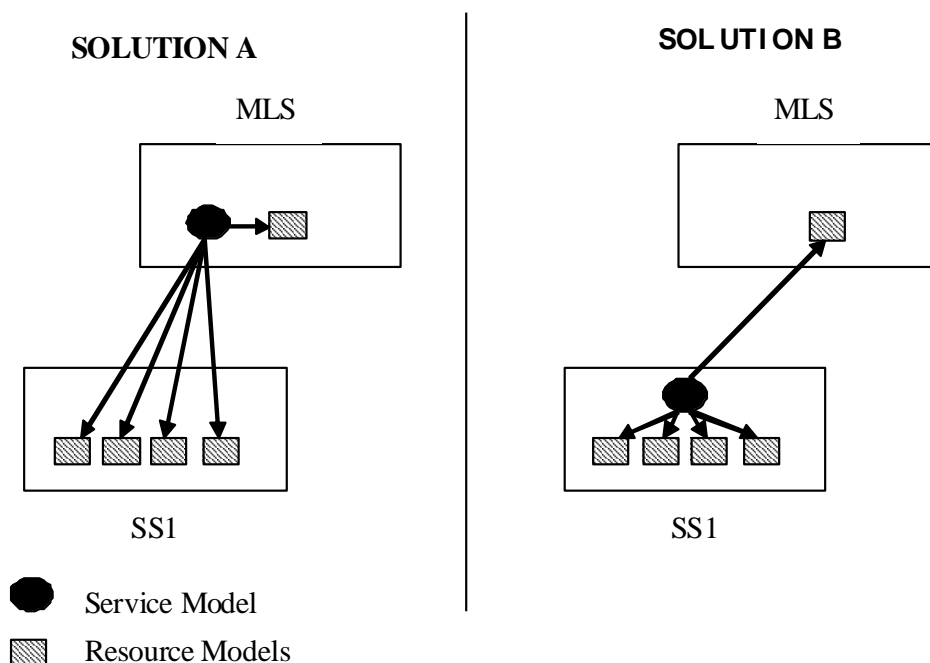
In addition to understanding which configurations are supported it is also important to consider the efficiency of the service model, and resource distribution impact. Monitoring a resource which resides in the same landscape is more efficient than monitoring a resource which resides in another landscape. When creating service models, strive to minimize the number of service resources which reside on remote landscapes. Consider the following example. Service XYZ has five resources, 1 of these resource models resides on the MLS while the other 4 reside on SpectroSERVER 1 (SS1). The following image depicts two potential designs for service XYZ:

### Service Model Example: Resource Distribution Impact

Because of the reduced number of resources residing on a remote landscape, Solution B is a more efficient design. If the service can reside on either landscape, select the landscape where most resources exist. You can summarize this statement by saying 'build the service closest to the bulk of its resources'.

There are some circumstances where the service has resources on multiple second tier landscapes and must reside on the MLS. In these scenarios, a more efficient service design can be created by consolidating the remote resources into a sub-service which is monitored by the parent service on the MLS.

The following image depicts two solutions which can be created when the service must reside on the MLS:

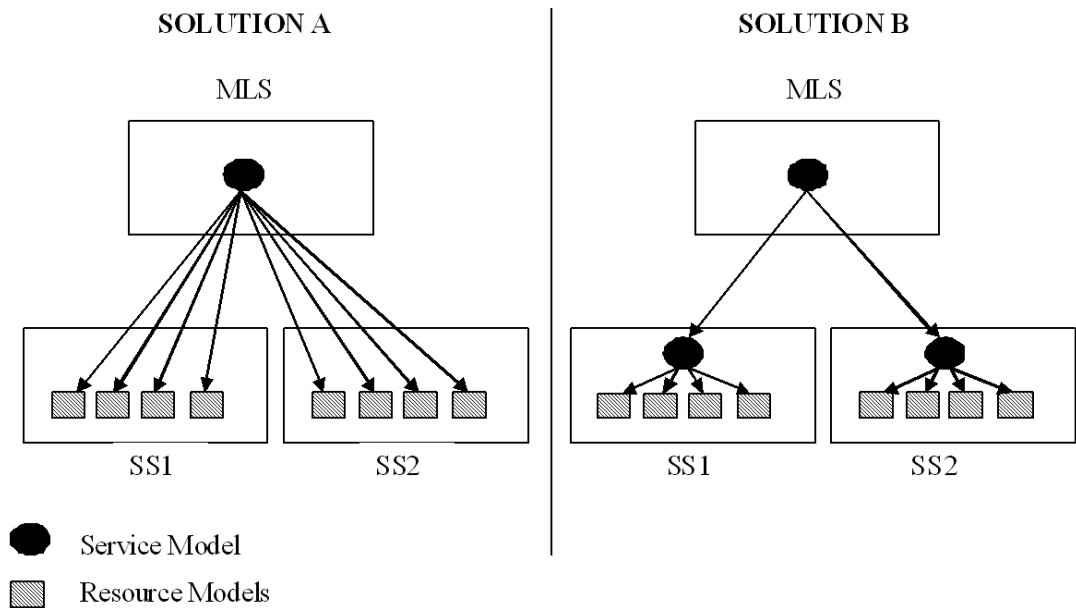


### Service Model Example: Service Resides on the MLS

Solution B represents a more efficient design by minimizing the number of resources which are monitored remotely. In Solution B, the service models that are created on SS1 and SS2 become resources of the service that is created on the MLS. When creating service hierarchies, such as the one depicted in Solution B, it is important to verify the policies that are used for each service, reflects the behavior of the resources from each landscape.

When creating services which use global collection consider the number of remote resources that result from the use of the global collection. When any type of container is used to specify service resources by default, the service monitors the contents of the container. A global collection can contain models from multiple landscapes.

The following image illustrates the use of a global collection that can potentially create an inefficient service design.



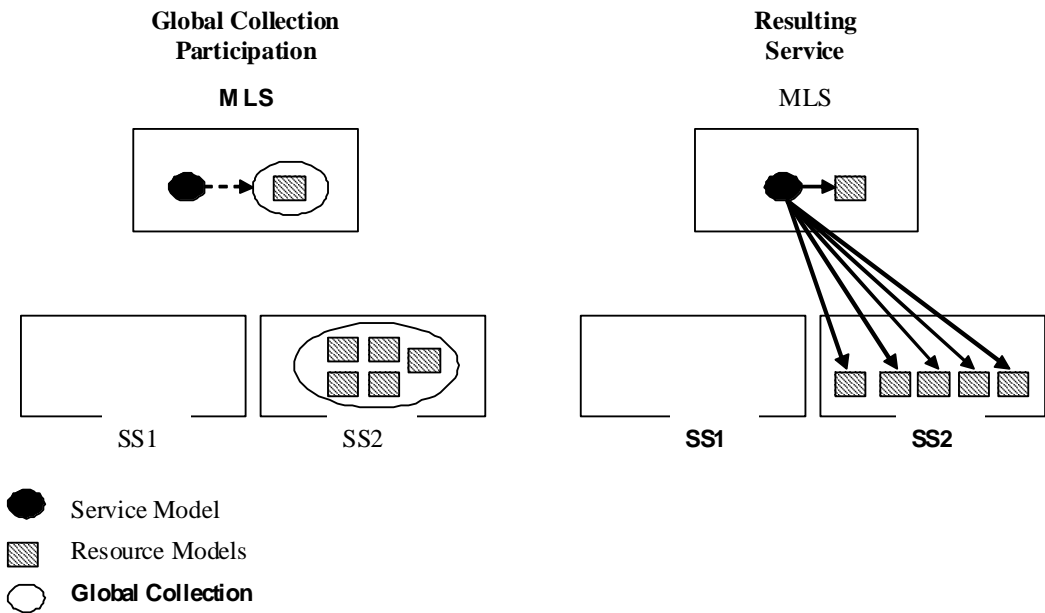
Again Solution B, represents a more efficient design by minimizing the number of resources which are monitored remotely. In Solution B, the service models that are created on SS1 and SS2 become resources of the service that is created on the MLS. When creating service hierarchies, such as the one depicted in Solution B, it is important to verify the policies that are used for each service, correctly reflects the behavior of the resources from each landscape.

When creating services which use Global Collection consider the number of remote resources that result from the use of the Global Collection. When any type of container is used to specify service resources by default, the service monitors the contents of the container. A Global Collection can contain models from multiple landscapes. Consider the following images, and how the use of a Global Collection could potentially create an inefficient service design.

### Example: Global Collection

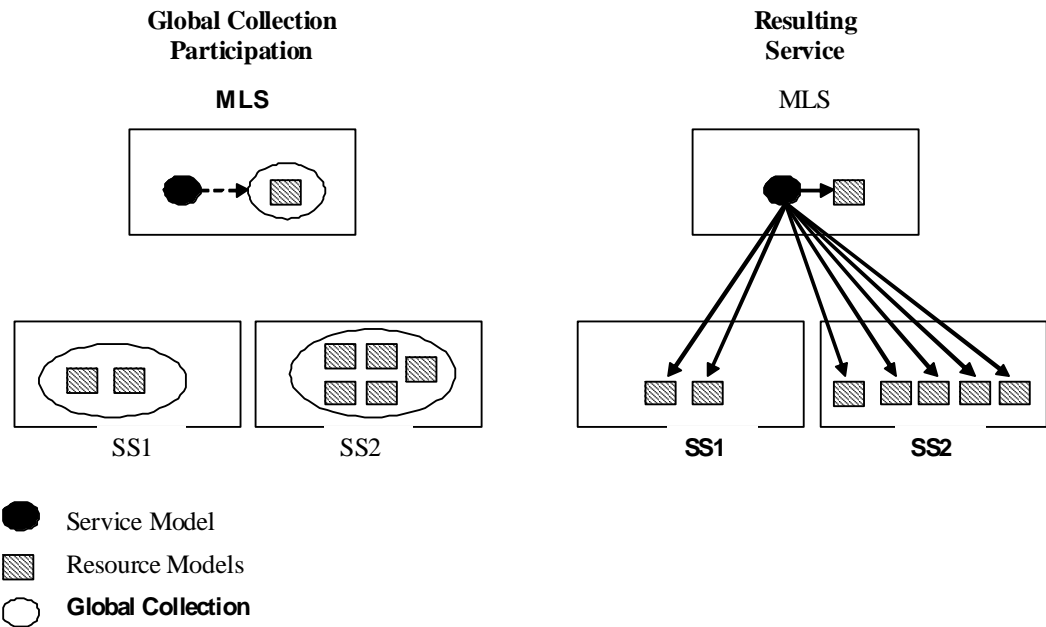
In the following scenario, if the service is created on the MLS it would be monitoring five remote resources. If the service was created on SS2, it only has to maintain one remote resource.

However, improving efficiency on this scenario cannot be as simple as moving the service to SS2. The service is using a global collection to specify its resources to support a set of potentially dynamic resources. Consider if two more models are created on SS1 which participate in the global collection.



### Global Collection Example: Add Two Additional Models

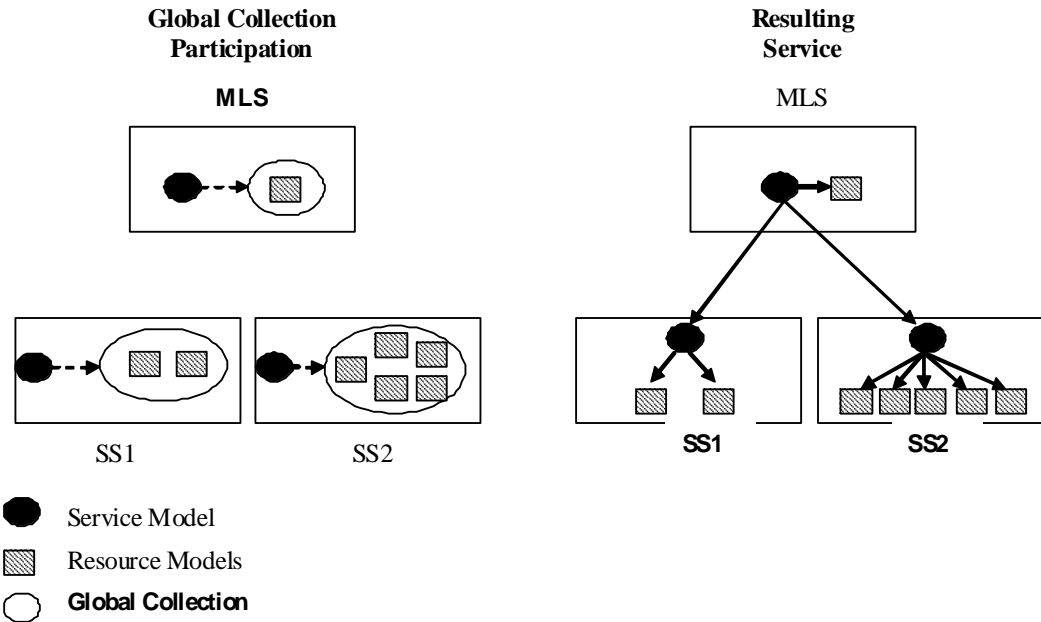
If the global collection contained resources from all landscapes, the service has to reside on the MLS. An alternative to consider is the use of multiple services which monitor the local copy of the global collection, residing on the same landscape. Although, logically a global collection is a single model, it is implemented as a set of duplicate models with a model residing on each landscape for which the global collection is specified.



### Example: Multiple Services Monitoring the Local Copy of the Global Collection

Although this approach does produce a more efficient service, it can be complex to maintain and may require the use of CA Spectrum Command Line Interface (CLI) to implement.

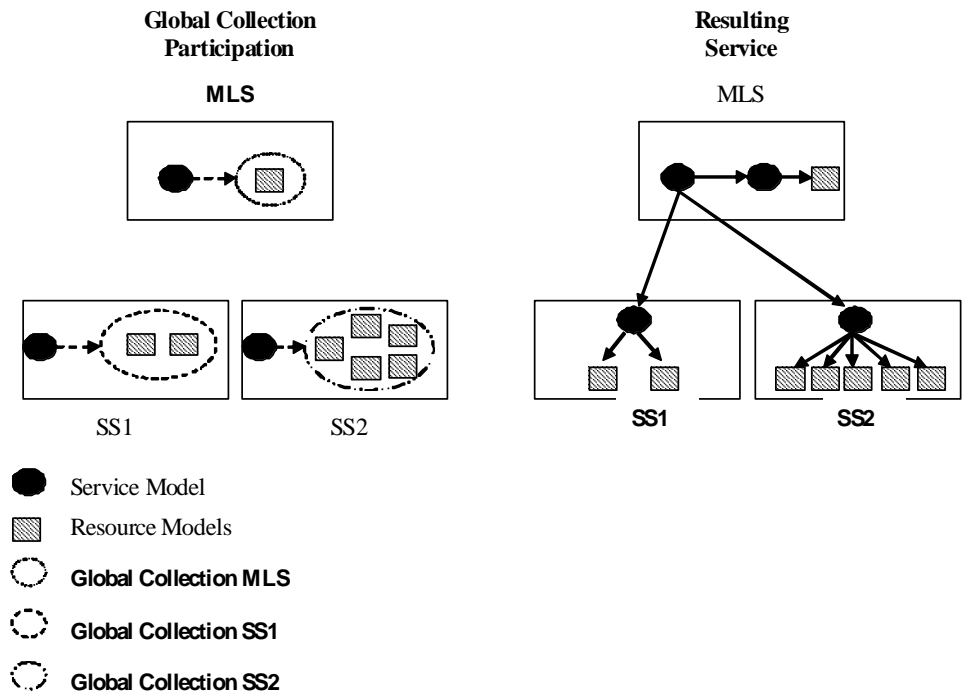
An alternate technique would be to create multiple global collections which are bound to a single landscape. Services can then be created on each landscape, with a parent service to monitor them.





Example: Multiple Global Collections Bound to a Single Landscape

This design allows for the benefits of a dynamic collection, but also provides an efficient service design. It may not be ideal to maintain multiple Global Collections in this way so you must weigh the costs and benefits of this solution.





# Chapter 3: Working with Policies and Policy Components

---

This section contains the following topics:

[Policies](#) (see page 75)

[Policy Types](#) (see page 76)

[Attribute Maps](#) (see page 81)

[Rule Sets](#) (see page 85)

## Policies

A service policy reflects the behavior of a set of resources that logically impacts a service. The policy specifies which resource attribute is monitored and how those attributes are interpreted to determine the health of a service. A number of common policies are available out-of-the-box, and users can create their own policies to more accurately monitor service resources.

A policy includes the following basic components:

### **Attribute map**

The attribute map serves two purposes. First, it specifies which resource attribute is monitored. Second, it maps resource attribute values to a set of resource health values. The mapping is done based on the logical severity of the attribute value. For example, if the attribute map represents the Port Status attribute, a disabled port can be logically considered as a down resource. If the attribute map represents the status of a response time test, a minor threshold violation and a slightly degraded resource can be considered.

This mapping allows policy rule sets to handle various resource types in a common way by having only to consider the resource health values of down, degraded, and slightly degraded.

### Rule set

A rule set consists of a number of statements evaluating the cumulative mapped resource health values of a set of service resources against some criteria. Each rule within the set specifies the criteria and the resulting service health value if the criteria are met. For example, a rule can look something like: When all resources are down the service is down. This means that given the mapping of the monitored attribute if all resources have a resource health of down the resulting health of a service using this policy can be down.

As mentioned, the rule set consists of multiple statements or rules. The rules are evaluated from the top down, and the first rule which is satisfied determines the health of any service using that policy. If none of the rules in the rule set are satisfied, the service health can be up.

**Important!** Editing and deleting policies and attribute maps and rule sets can be performed by authorized personnel who are aware of the potential ramifications of these actions, particularly policies for services that are watched by Service Level Agreements (SLAs).

## Policy Types

Before you create a custom policy, have a clear understanding of what you want to monitor (the watched attribute and its status values) and the method that is used for the monitoring. Two distinct types of service monitor policies are available, such as:

- Status policies use an attribute map to compare the status of monitored attributes that are indicated by the rule sets. These policies use the *All*, *Any*, or *Percentage* rule sets.
- Statistical policies use Aggregate rules and compare the value of the monitored attribute to the computed value set by each rule. A statistical policy requires an attribute map to specify which attribute can be monitored, but the mapped values are ignored in favor of the attributes pure values. The attributes values can be summarized in several ways, such as average, minimum, maximum. The summarized values are then compared against numeric thresholds that are specified in the rule-set.

For example, a status policy can monitor the operational status of an interface, where a statistical policy can monitor the error rate of an interface.

## Create a Policy

You can create policies using any combination of standard and user-defined attribute maps and rule sets. When you create a policy, it becomes available to other Service Manager users. You can also create a policy by saving a uniquely named version of an existing policy.

### Follow these steps:

1. [Open the Service Policy Editor](#) (see page 19).

2. Click the Policies tab and click Create.

The Create Policy dialog appears. Service Manager autofills the Author field with your CA Spectrum user name.

3. Enter a unique name for the policy in the Policy field.

4. Select an attribute map.

Properties for the map you select appear. If the attribute maps available do not meet your requirements, you can [create an attribute map](#) (see page 82) or can edit [an existing user-created attribute map](#) (see page 84).

5. Select a rule set.

Rules for the rule set you select appear. The rule set specifies the status (or health) of a service is based on the mapped health values of the service resources. A rule set consists of conditional statements that assert what the health of the service is based on the collective set of resource health values.

If the available rule sets do not meet your requirements, you can [create a rule set](#) (see page 87) or can [edit an existing user](#) (see page 88).

6. (Optional, and only applicable if you chose a Condition attribute map).

Specify alarm type exemptions for the policy. This option consists of a set of alarm types and how they are applied either inclusively (caused by) or exclusively (not caused by).

You can specify alarm type exemptions at the policy level, if the policy is used by multiple services. If the alarm type exemption configuration is only relevant for a single service, it can be better to define it for the service itself rather than at the policy level.

Consider the following points before you allow or disallow alarm type lists in policies:

- Alarm type exemptions that are configured at the policy are enforced only for services using the policy which do not define their own alarm type exemptions. If a service using the policy is later edited and has alarm type exemptions that are configured for it, the policies configuration can be ignored from that point forward.
- Any changes that you make to the alarm type exemption configuration for a policy affect all services that use the policy.

Follow these steps to specify alarm type exceptions for a policy:

- a. In the policy create/edit dialog Click Set. (If the set button is disabled, it is likely that the specified attribute map is not for the Condition attribute, only Condition-based policies can specify alarm type exemptions).

The Configure Inclusive or Exclusive Alarm Types dialog appears.

- b. Specify Caused By or Not Caused By.
- c. Select the appropriate alarm types and click OK.
- d. The Create Policy dialog shows the alarm type inclusion or exclusion list that you specified.

7. Click Create.

The policy appears in the Policy list in the Service Policy Editor dialog.

**More information:**

[Add Alarm Types to a Custom Condition Policy](#) (see page 78)

## Add Alarm Types to a Custom Condition Policy

You can add resource impacting alarm to the list of alarm types that are specified in any user created policy which uses a Condition attribute. This option is available when selecting an Alarm in the Alarm tab of the OneClick Contents Panel.

**Follow these steps:**

1. Select one or more alarm types to include in the policies from any OneClick alarm view.
2. Right-click and click Add Alarm Type to Service Policies.

The Specify Alarm Types for Service Policies dialog appears. This dialog lists selected alarm types and all custom (user-created) Condition policies.

**Note:** You can include the Alarm Behavior column by right-clicking any table column heading. This indicates how the policy interprets the alarm exemption Caused By or Not Caused By.

3. Select the policies to which you want to add the alarm types and click OK.

The alarm types are included in the alarm type list for each selected policy. If you selected a policy without an alarm type configuration, Service Manager defaults to a setting of Not Caused By for the policy.

## Create a Policy from a Copy

You can create a policy by copying an existing policy and saving it with the settings you require under a different name.

**Follow these steps:**

1. [Open the Service Policy Editor](#) (see page 19).
2. Click the Policies tab, select the policy from which you want to create a policy, and click Copy.

The Create Policy: <Policy Name> dialog appears. It includes the settings for the policy you are going to use as the basis for your new policy.

3. Enter a unique name for the new policy in the Policy field.
4. Edit policy properties, as described in [Create a Policy](#) (see page 77), and click Create.

The new policy appears in the Policy list in the Service Policy Editor dialog.

## Edit a Policy

You can modify all user-created policies. Service Manager implements modified policy settings immediately after you save them.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Policies tab, select the policy that you want to edit, and then click Edit.

The Edit Policy: <Policy Name> dialog appears.

**Note:** Policy edits are applied to all services or resource monitors which are currently using the policy. Depending on the nature of the edit, it results in a change to the service health of one or more services.

3. Edit the settings, as described in [Create a Policy](#) (see page 77), and click OK.

The policy is edited.

The following limitations apply to the service policy editing:

- You can edit or delete user-created policies, attribute maps, or rule sets. However, Service Manager prevents you from deleting any policy (or any attribute map or rule set that are part of it) that is used by a service or resource monitor.
- You cannot edit or delete CA-authored Service Manager policies.
- You can copy any policy and can save it under a unique name and can edit it or can delete.
- You cannot edit or delete CA-authored attribute maps and rule sets.  
You can copy any attribute map or rule set and can save it under a unique name.

## Delete a Policy

You can delete any user-created policy that is not currently in use. If the policy used by a service deletion fails, an error dialog indicating the failure is displayed.

**Note:** When you delete a policy, do not delete its attribute map and rule set.

**Follow these steps:**

1. [Open the Service Editor dialog](#) (see page 18).
2. Click the Policies tab and then select the policy that you want to delete.
3. Click Delete.

The policy is removed from the Policy list.



## Attribute Maps

Attribute maps associate natural resource attribute values to equivalent resource health values, and let you specify a root cause reason string for each mapped value. Any attribute which has whole number values can be mapped. Multiple attribute maps can exist for the same attribute, and can have different value mappings or simply different root cause reason strings. Each attribute map has its own name which describes its purpose.

The attribute map serves two purposes when added to a service policy. First it specifies which resource attributes monitored. It provides a mapping of how the resource attribute values indicate the health of the resource. When the specified attribute changes for a resource model, the service evaluates the value to determine its relative resource health and apply that health value with all of the other resources health values to the rule set.

If the out-of-the-box attribute maps are not sufficient, you can create a custom attribute map. All user-created attribute maps can be edited and deleted by any Service Manager user.

Consider the following information about attribute maps:

- An attribute map must have a unique name.
- You cannot edit or delete CA-authored attribute maps.
- Any attribute can have multiple mappings.
- Service resource(s) must support the attribute that is specified in the attribute map. CA Spectrum generates a minor (yellow) alarm on the Service Management model, if a service uses a policy that specifies a watched attribute that is not supported by the service resources.
- You must know what values an attribute returns and what they indicate about the state of the resource(s) on which they are monitored to map those values to service health values.
- At a minimum, an attribute map must include one mapped value, and must define a default root cause reason.
- You can map multiple attribute values to the same service health value. For example, port statue values of disabled, down, and unreachable are all mapped to a resource health of down.
- If the attribute you select returns enumerated values, each value must be mapped to up, down, degraded, or slightly degraded before you can save the new mapping.

- Service Manager attribute maps now support range values in addition to single value mapping.

For example, an attribute map value can appear in this form single value 100, range value 100-200. Using the following information, you can configure attribute maps:

- 1-99 = Slightly Degraded
- 100-199 = Degraded
- 200-300 = Down

Also supported is greater than and less than operators, so a set of mapped values may look like the following equations:

- <100 = Down
- 100-200 = Degraded
- 300-400 = Degraded
- >400 = Down

## Create an Attribute Map

You can create a custom attribute map using Service Policy Editor.

### Follow these steps:

1. [Open the Service Policy Editor](#) (see page 19).
2. Click the Attribute Maps tab and click Create.

The Create Attribute Map appears.

3. Specify the following properties:

#### **Attribute Map**

Identifies the attribute map. Provide a unique name.

#### **Default Root Cause Reason**

Identifies the underlying reason for the service health state. Later, when you map particular attribute values to service health values, you can overwrite the default reason for each mapping.

4. Click Attribute.

The Attribute Selector dialog appears.

5. Select an attribute of the allowed type (counter, integer, date, time ticks, and gauge), and then click OK.

The Create Attribute Map dialog appears.

6. Click Add.

The Add Value to Service Health Mapping dialog appears.

7. Enter the following values:

**Attribute Value**

Specifies the attribute value that you want to map to a service health value.

**Service Health**

Specifies the service health value that you want to map to the attribute value.

**Root Cause Reason**

Describes the root cause that you want to provide for this particular mapping. The root cause reason is displayed in the root cause tab of the OneClick Component detail panel, and the Outage Details in a service outage history table.

You can create new attribute maps specifically to provide better root cause reasons. For example, the standard Condition attribute map defines generic root cause reasons.

It is useful to create multiple purpose-specific attribute maps for the Condition attribute. For example, you can define a Condition attribute map specific for services monitoring cable modems and can define root cause reasons like: "Modem disconnect, and Data transfer fault." These attribute maps are more descriptive strings, which are used in place of standard Condition strings of "A critical problem was detected on the resource, and A major problem was detected on the resource."

8. Click OK.

The Mapped Values panel lists the new attribute value and service health mapping.

9. Click Create to save the new attribute map.

The new attribute map appears in all attribute map lists in Service Manager, and it can be used in all user-created policies.

**More information:**

[Create a Policy](#) (see page 77)

## Create an Attribute Map from a Copy

You can create an attribute map by copying an existing map and saving it with the settings you require under a different name.

**Follow these steps:**

1. [Open the Service Policy Editor](#) (see page 19).
2. Click the Attribute Maps tab, select the map that you want to copy, and click Copy.  
The Create Attribute Map dialog appears. It includes the settings for the map you want to copy.
3. Specify a unique name, modify settings as required (using the set command where applicable), and click Create.

The new attribute map appears in all attribute map lists in Service Manager, and it can be used in all user-created policies.

## Edit an Attribute Map

You can edit any user-defined attribute map regardless of whether it is included in a policy that is in use. If edits to the attribute map are saved, all services or resource monitors using related policies reevaluates their health that is based on the edit and can result in a change in service health for the models.

Consider the following information before editing an attribute map:

- You can change the attribute map name, the service health designation for an attribute value, the default root cause reason, and the root cause reason for each service health mapping.
- You cannot change the attribute that is originally specified.

**Follow these steps:**

1. [Open the Service Policy Editor](#) (see page 19).
2. Click the Attribute Maps tab, select the attribute map that you want to edit, and then click Edit.

The Edit Attribute Map: <Attribute Map Name> dialog appears.

3. Modify settings as required (using the set command where applicable), and click OK.

The attribute map is edited.

**More information:**

[Create a Policy](#) (see page 77)

## Delete an Attribute Map

You can delete any user-created attribute map that is not part of a policy currently in use.

**Follow these steps:**

1. [Open the Service Policy Editor](#) (see page 19).
2. Click the Attribute Maps tab, select the map that you want to delete, and click Delete.

The attribute map is removed from the attribute maps list.

## Rule Sets

A rule set consists of a set of rules. Each rule is a conditional statement that is comprised of a comparator and a resulting health value. A rule is considered satisfied if the cumulative resource health matches its criteria. For example, when all resources are Down the service is Down. If all resources within the service have a resource health of down, the rule can be satisfied, and the policy can be evaluated to a resulting health of down.

Rules are evaluated from top down, the first rule within the rule set which is satisfied dictates the health of any service or resource monitor using the policy. It is important to consider rule evaluation when creating a rule set, and verify that rules of lesser significance do not hide rules of greater significance. For example, consider a rule set with the following logic:

Rule 1 = When any 1 resource is Down the service is Degraded.

Rule 2 = When all resources are Down the service is Down.

A rule set like Rule 1 and Rule 2 can never return anything other than Degraded, because even if all resources are down, Rule 1 is still satisfied as one resource is down.

Service manager supports four different categories of rules which can be used to form a rule set. All, Any, and Percent Rules use the mapped values that are provided by the attribute map. Aggregate rules use the resource attributes natural values.

Verify the following rule types:

### All

When all the monitored service resources or all resources that are watched by a resource monitor have the service health value (down, degraded, or slightly degraded), the service or resource monitor is (down, degraded, or slightly degraded).

### **Any**

When a particular number of monitored service resources or a particular number of resources that are watched by a resource monitor have the service health value (down, degraded, or slightly degraded), the service or resource monitor is (down, degraded, or slightly degraded).

### **Percent**

When the percentage of monitored service resources or the percentage of resources that are watched by a resource monitor have the value (down, degraded, or slightly degraded), the service or resource monitor is (down, degraded, or slightly degraded).

### **Aggregate**

When the (Sum, Minimum, Maximum, and Average) for all monitored service resources or the (Sum, Minimum, Maximum, and Average) of resources that are watched by a resource monitor is (less than, greater than, or equal to) the Integer or attribute value you designate, the service or resource monitor is (down, degraded, or slightly degraded).

A status policy rule set can consist of any combination of All, Any, and Percentage rules. Because mapped values are ignored, a statistical rule set can use only Aggregate rules.

Consider the following information about rule sets:

- You can create any number of uniquely named rule sets.
- Uniquely named rule sets can include identical rules.
- You can create new versions of existing rule sets.
- You cannot edit or delete CA-authored rule sets.
- The order in which you list a rule in rules set is important. The first rule that is satisfied dictates the service health value returned.

## Create a Rule Set

You can create an original custom rule set or another version of an existing rule set in the Service Policy Editor if the CA-authored rule sets do not meet your requirements for a policy. All user-created rule sets can be modified and deleted by any Service Manager user.

**Follow these steps:**

1. [Open the Service Policy Editor](#) (see page 19).
2. Click the Rule Sets tab and click Create.  
The Create Rule Set dialog appears.
3. Enter a name for the new rule set in the Rule Set field.  
The Author field autofills with the current Service Manager user name.
4. Click Add to create a rule for the rule set.  
The Create Rule dialog appears.
5. Configure rule parameters, including the type and the conditions for the type, and click OK.  
The rule appears in the Create Rule Set dialog.
6. Rearrange rules as necessary using the Up Arrow and Down Arrow buttons, and click Create.  
The rule set is created.

**More information:**

[Create a Policy](#) (see page 77)

## Create a Rule Set from a Copy

You can create a rule set by copying an existing rule set and saving it with the settings you require under a different name.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 19).
2. Click the Rule Sets tab, select the rule set you want to copy, and click Copy.  
The Create Rule Set dialog appears. This dialog includes the settings for the rule set you are going to use as the basis for your new rule set.
3. Specify a unique name and edit rules and modify rules as required, and then click Create.  
The rule set appears in all rule set lists in Service Manager, and it can be used in all user-created policies.

## Edit a Rule Set

You can edit any user-defined rule set regardless of whether it is included in a policy that is in use. If edits to the rule set are saved, all services or resource monitors using related policies reevaluates their health that is based on the edit, that results in a change in service health for the models.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Rule Sets tab, select the rule set you want to edit, and then click Edit.  
The Edit Rule Set: <Rule Set Name> dialog appears.
3. Modify the name of the rule set in the Rule Set field.
4. (Optional) Use the arrow keys to rearrange rules.
5. Edit a rule by selecting the rule and clicking Edit.  
The Edit Rule dialog appears.
6. Change rule settings as needed and click OK.  
The Edit Rule dialog appears.
7. Click OK.  
Your edits are saved.

**More information:**

[Create a Policy](#) (see page 77)



## Delete a Rule Set

You can delete a rule set which is not part of a policy currently in use by a service or resource monitor.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Rule Sets tab view, select the rule set you want to delete, and click Delete.

The rule set is removed from the rule sets list.



# Chapter 4: Creating and Managing Customers

---

This section contains the following topics:

[Customers and Customer Groups](#) (see page 91)

[Edit Customer Settings](#) (see page 93)

[Edit a Customer Group](#) (see page 94)

[Move a Customer or a Customer Group](#) (see page 94)

[Delete a Customer or a Customer Group](#) (see page 95)

[Associate a Service or an SLA with a Customer](#) (see page 95)

## Customers and Customer Groups

Customers are CA Spectrum models that represent a person or organization that is associated with services or SLAs. The use of customer models enables you to track and monitor each customer service and SLAs.

A customer model status attribute reflects the status of the customers services. The customer status can be equivalent to the worst service health value for all of the customer services. For example, assume that a customer is associated with services A, B and C. If service A is up, service B is degraded, and service C is down, the customer status attribute has a value of severe to reflect a service down. If service C is restored to up and B remains degraded, the customer status attribute indicates significant impact to reflect a service degraded. Customer icons within OneClick and the Service Dashboard indicate the value of the customer status attribute. No alarms are associated to changes in customer status. The visual indication is only the icon color.

Each customer model also has a criticality attribute with values ranging from low to high. Similar to the criticality of a service model, all or a portion of the customer models criticality is added to the impact of any resource alarms that are affecting the customers services. This confirms that alarms impacting highly critical customers has a high impact value.

Customer models can be added to customer groups. The customer group model provides a way to organize similar customers. The condition of a customer group can be equivalent to the worst status of all customers within the group. No alarms are associated to condition changes for the customer group model, the only visual indication is icon color.

Customers and customer groups you create appear under the Customers tab in the Service Editor, in Service Dashboard, and in OneClick under Service Management in the Navigation panel.

## Create a Customer

A customer identifies a person or an organization that is associated with a service, an SLA, or both. In addition to the Criticality and Security String parameters included for all CA Spectrum models, it includes customer identity and contact information and other fields in which you can enter additional information.

### Follow these steps:

1. [Open the Service Editor](#) (see page 18).
2. Click the Customers tab.  
A list of customers that are created in Service Manager appears.
3. Click Create Customer.  
The Create Customer dialog appears.
4. Complete the required fields (denoted by asterisks).
5. (Optional) Specify the customer group (using the Customer Group tab) to which you want to add the customer and click Create.  
The customer is created.

### More information:

[Example: Define a Customer and a Customer Group](#) (see page 137)

## Customer Criticality and a Service's Outage Alarm Priority

A customer criticality is factored into the impact calculation of any alarms that cause a service outage for one of the customer services. For example, Customer A has a Medium criticality value of 15 and Customer B has a Low criticality value of 5 and both Customer A and B are associated to services that have high criticality values and are down.

The root cause alarm for the outage on Customer A service has an increased impact of + 30 for the service and + 15 for the customer. The root cause alarm for Customer B service has an increased impact of +10 for the service and +5 for the customer.

Because Customer A has a higher criticality value, the alarms that affect Customer A service have a higher impact. Although both example alarms affect high criticality services, the root-cause alarm that affects Customer A service has a higher impact values. If the alarm view in OneClick is ordered by alarm impact, the alarm affecting Customer A can appear higher in the alarm table.

## Create a Customer Group

Service Manager lets you organize customers in groups in any way that meets your requirements for tracking and managing customers.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Customers tab and click Create Group.  
The Create Customer Group dialog appears.
3. Enter a Customer Group Name, the landscape where you want to create the group, and, optionally, a security string.
4. Under Group Location, select the group in which you want the new customer group saved.

**Note:** By default, Service Manager saves all customers and customer groups under the customer manager model on the landscapes where the customer model or customer group model exists.

5. Click Create.  
The customer group is created.

**More information:**

[Example: Define a Customer and a Customer Group](#) (see page 137)

## Edit Customer Settings

You can edit customer settings as required.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Customers tab, select the customer that you want to edit, and then click Edit.  
The Edit Customer dialog appears.
3. Modify the settings:
  - Under the Contact Information tab, you can edit all contact information except the landscape where the customer was created.
  - Under the Customer Group tab, you can move the customer to a new location.Click OK.  
The customer settings are edited.

## Edit a Customer Group

You can edit customer group settings as required.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Customers tab, select the customer group that you want to edit, and click Edit.

The Edit Customer Group dialog appears.

3. Modify the settings.

**Note:** You can modify the group name and security string. You cannot modify the landscape where the group was created.

4. Click OK.

The customer group is edited.

## Move a Customer or a Customer Group

As your customer list grows, you can reorganize your customers and customer groups by moving them from their current locations to new locations. When you move a customer group, you also move its customers with it.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Customers tab, select the customer or customer group you want to move.
3. Drag and drop the customer or customer group to the new location.

The customer or customer group is moved.

## Delete a Customer or a Customer Group

You can delete customers and customer groups you no longer use. When you delete a customer group, you can select to delete or retain its customers.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Customers tab, select the customer or customer group you want to delete, and then click Delete.

**Note:** If you are deleting a customer group, you are prompted to retain the group customers or delete them with the group.

3. Respond to the confirmation message that appears to complete the deletion.

The customer or customer group is deleted.

## Associate a Service or an SLA with a Customer

Service Manager lets you associate customers with services and SLAs. Verify the following benefits:

- You can track services and SLAs associated with customers in Service Dashboard, Service Editor, and OneClick
- You can generate service and SLA reports about specific customers with CA Spectrum Report Manager.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the Customers tab, select the customer that you want to associate with a service or SLA.
3. Click the Services or SLAs tab and click Select Customer Services or Select Customer SLAs.

The Select Services or Select SLAs dialog appears.

4. Move the services or SLAs you want to associate with the customer from the Available Services or Available SLAs list to the Customer Services or Customer SLAs list (Do the opposite to remove services or SLAs).

5. Click Ok,

The service or SLA is associated with the customer.

**Note:** Only the real-time health of services impact the status of associated customers. The customer status attribute indicates the real-time status of a customer in terms of any impacted services. Changes to the SLA status of SLA models associated to a customer does not alter the customer models status as the SLA status does not indicate a real-time value.



# Chapter 5: Creating and Managing Service Level Agreements

---

This section contains the following topics:

- [About Service Level Agreements](#) (see page 97)
- [Create an SLA](#) (see page 101)
- [Create an SLA From an SLA Template](#) (see page 103)
- [Guarantee Types](#) (see page 105)
- [Create a Guarantee for a Top-Level Service](#) (see page 106)
- [Create a Guarantee for a Service, Sub-Service, or Resource Monitor](#) (see page 107)
- [Edit a Guarantee](#) (see page 110)
- [Delete a Guarantee](#) (see page 111)
- [Create an SLA Period](#) (see page 111)
- [Edit an SLA](#) (see page 112)
- [Delete an SLA](#) (see page 113)
- [Associate a Customer with an SLA](#) (see page 113)
- [SLA Templates](#) (see page 114)
- [Guarantee Templates](#) (see page 116)

## About Service Level Agreements

CA Spectrum represents a service level agreement or operational level agreement with an SLA model. A Service Manager SLA model or SLA, incorporates measurable provisions which are defined by the service or operational level agreement. These provisions are implemented as service models within CA Spectrum which are in turn monitored by the SLA model. The SLA monitors the real-time health of each associated service, and records outage time when the service is down. The recorded time is compared against a number of thresholds to determine the status of the SLA for a given period. SLA models can be created as individual SLAs or from preconfigured SLA Templates.

## Guarantees

Each SLA specifies one or more guarantees. A guarantee is a CA Spectrum model that represents and monitors a provision of the SLA. Each guarantee is associated to a service or resource monitor. The guarantees record service outage time and compares the amount of recorded time to a threshold specified by the user. A guarantee is considered violated if the amount of accumulated service down time exceeds the threshold that is specified by the users. If the guarantee has not recorded any outage time, or the recorded outage time is less than the threshold, the guarantee is considered compliant. The status of an SLA equals to the status of its worst guarantee. In other words if any one of an SLAs guarantee is violated, the SLA is violated. The status of an SLA is always expressed in terms of a period. Once an SLA is violated, it remains violated during the current period, unless outages contributing to the violation are edited in such a way that recorded outage time is removed from the guarantee.

Many SLAs include some form of an availability guarantee. For example, an SLA can stipulate that the service is guaranteed to be available 99.9 percent of each month. These statements define both the availability threshold and the period for which the threshold applies.

SLAs commonly include performance or response time provisions. For example, an SLA can state that service response time is 100 ms or better 99.9 percent during the hours of 8 AM to 5 PM each weekday.

Service Manager provides both Availability and Response Time guarantees. Functionally availability and response time guarantees are similar in the way that outage time is recorded. The Availability guarantees offer three additional thresholds for mean time to repair, such as maximum outage time and mean time between failures. The distinction of availability that is compared to response time is used for organizational convenience rather than functional purposes, as an availability guarantee can be configured to perform identically to a response time guarantee.

An SLA can include as many guarantees (either availability or response time) as required, to monitor all measurable provisions in a service level agreement. The SLA is considered compliant when all of its guarantees are compliant, or violated if any of its guarantees are violated. Guarantees and SLAs can also have a status of warned or at risk if a guarantee has record service outage time such that a violation is likely to occur. When the status of the SLA changes to a warned or violated state, an alarm is generated on the SLA model. This alarm remains on the model until the SLA period ends, or a user initiated outage edit causes the recorded time to fall below the threshold.

Guarantee thresholds can be configured in one of two ways. First and most common is by percentage of availability. When configuring percentage-based thresholds a user specifies the desired availability for the guaranteed service. User can also configure guarantee thresholds by the number of seconds of outage time not to be exceeded for the period. Configuring a threshold that is based on a set number of seconds can eliminate some of the variability found with percentage-based thresholds for monthly periods.

Regardless of how the threshold is configured, the guarantee determines its status that is based on the amount of recorded outage time compared to the amount of allowable outage time for the period. Consider an SLA that stipulates that a service must be available 99.5 percent of the time for the SLA period. The SLA must include a guarantee that specifies the availability threshold (99.5 percent). Stated another way, the service cannot be down more than 0.5 percent of the total time for the SLA period, in this case a calendar month. For a thirty-day month, out of 720 available hours, the service cannot be down more than 3.6 hours.

In addition, the SLA stipulates that no individual service outage can exceed 15 minutes, and the average time to repair an outage cannot exceed 10 minutes. These statements indicate that one addition to overall threshold of 99.5 percent, the guarantee can also specify threshold for MOT (maximum outage time), and MTTR (mean time to repair). If any of the guarantees thresholds are violated the guarantee, and likewise the SLA is considered violated for the period.

## Period

A *SLA period* is the interval for which the guarantees can compute their status. The most common SLA periods are monthly. The guarantees that are specified for the SLA records time and calculates status on a monthly basis, with a specific start and end time. For example, if the SLA period is monthly, the first period begins at midnight on the first of the month, and end at midnight on the first day of the following month. The next SLA period begins immediately after the current SLA period ends.

In addition to the overall SLA period, some service level agreements can define specific hours that a guarantee is relevant for. For example, the SLA may state availability of 99.9 percent from 8 AM to 5 PM, Monday through Friday. These time frames within the period are named *business hours*.

*Business hours* can also be described as the times when the guarantee is active. A guarantee does not record outage time for service outages occurring outside of its business hours. Percentage-based guarantee thresholds are also calculated specifically to include only the time for which the guarantee is active. Therefore, if a guarantee defines a percentage-based threshold then the business hours has far less allowable outage time than a guarantee that specifies the same threshold, but no business hours.

## SLA Considerations

It is important that SLA status is accurate based on the guarantees stipulations in the SLA. In most of the cases, you can notice that not all service outages can impact the SLA. One of the most common problems that users can encounter when using SLA models is guarantees recording outage time for issues that are not guaranteed by the SLA. The reason for this is that services designed for accurate real-time monitoring generally do not lend themselves well to SLAs.

When designing a service to encompass all possible fault scenarios, you can build a vast set of resource monitoring capability. SLAs on the other hand tend to be focused on specific types of service outages. For example, consider a critical application service. For real-time monitoring, you may want to understand the availability of the physical servers, the critical processes, system resources, network connections, application, and network response time. For accurate real-time management, it is necessary that the service considers a wide range of potential resource faults.

In support of the application service, the server team has an SLA that stipulates the physical servers, available 99.9 percent of the time during regular business hours. The server team is responsible for the physical servers, but not for the applications running on them, or the network which provides access to them. So, the availability guarantee for the server teams, SLA should only record time if there is a failure of the server itself, but not if there is a failure of the application, or a network failure that prevents access to the server.

Therefore, the design of a service that is designed for real-time monitoring is likely to be different from a service that is designed to isolated specific resource faults for an SLA. Let us look at a couple aspects of the previous scenario, and consider how service design can be different.

First consider how the availability of the physical servers is monitored in CA Spectrum. If contact is lost to the server, then the server can be considered not available. In terms of SLA; however, it is not enough to say that contact is lost, the SLA must consider the distinction between a server outage and a network outage preventing access to the server. In CA Spectrum, the availability guarantee must only record outage time if the server is red, and must not record outage time if the server is gray. If the application service is designed for real-time monitoring, it is unlikely that its designers included resource monitoring components to distinguish between these two types of outages.

Next consider how server faults must be distinguished from application or process failures. The server team is not responsible for the processing running on the server itself. To begin any monitored processes outages probably should be isolated to the process model and not the server. Other system resource-related failures may or may not be the responsibility of the server team, for example, high CPU is likely the result of the applications on the server, failures of local files systems on the other hand could be the responsibility of the server team. Again it is unlikely the original designers of a service for real-time monitoring would have considered making these distinctions in their service design.

You can use the two techniques to verify that SLAs accurately monitor the services they guarantee.

The first option is to simply create a parallel set of services which specifically monitor the components that are guaranteed by the SLA. These services are built in addition to services designed for real-time monitoring. They can monitor a smaller set of resources, possibly use different service policies, and does not generate alarms. SLA-specific services can also define periods of maintenance which makes them inactive outside of the SLA specified business hours. This is probably the easiest technique, but it does not always scale well. Consider the example of the server team SLA, imagine that the application team and the network also have SLAs. For one logical application service, you can easily produce four service hierarchies to support, one for real-time monitoring, one server-specific, one application specific, and one network specific. The implementation and maintenance of these services can be daunting.

The second option, an alternate technique, is to verify that all services are built with a modular design. It is valuable to remember that the use of resource monitors makes it easier to extend the monitoring capability of a service. This is true not only when adding new resources, but also when adding support for SLAs. Resource monitors are used to isolate specific faults such that guarantees can be associated to the specific resource monitor. If well-designed, the same service can be used for multiple SLAs, and still used for real-time monitoring.

## Create an SLA

When you create an SLA, name it, specify the period that it is in effect, and the guarantee thresholds for the services or resource monitors that are monitored by the SLA. After you create an SLA, associate it with an SLA customer. You can also modify an SLA as necessary when service delivery requirements change.

Consider the following points before creating an SLA:

- An SLA can have multiple guarantees.
- You can associate a single service to an SLA, but guarantees can be created for that service, or any of its sub services or resource monitors.
- You may need to create new service components to isolate the specific set of faults for which the guarantee is responsible.
- SLA models reside on the same landscape as the service the SLA is associated to.

### Follow these steps:

1. [Open the Service Editor](#) (see page 18).
2. Click the SLAs tab and click Create.

The Create SLA dialog appears.

3. Specify the following SLA properties:

#### SLA Name

Identifies the SLA model. OneClick lists SLA names under the Service Management - SLAs category in the Explorer view for each landscape where you have created SLAs. You can use duplicate names for SLAs. However, to facilitate filtering when searching through a lengthy list of identically named SLAs, provide different descriptions for each SLA.

#### Control

Specifies whether the SLA is activated during the current SLA period (Active), the default setting, or the next period (Inactive Until Next Period).

**Note:** If you activate an SLA during an SLA period, Service Manager does not prorate allowable outage time. The service that is associated with the SLA is allowed the amount of down or degraded time that is specified for the entire period. For example, if an SLA allows five hours of outage for a 30-day month and activate an SLA on 15th of the month, the service that is associated with the SLA can be unavailable for up to five hours over the course of remaining 15 days. In terms of an availability obligation, this situation allows for twice as much service outage than a service customer would expect over the remaining 15-day period. In this case, you can modify the availability threshold for the SLA guarantee(s) to an amount of time for the partial period proportional to the entire period. For example, in the previous example, you could change the availability threshold to two and half hours for the partial 15-day period.

#### Description

(Optional) Describes the SLA. You can enter unique descriptions for SLA that have the same name to facilitate finding each SLA using a list filtering capabilities.

#### Security String (Optional)

Identifies the security string for the SLA model. The security string secures access to the SLA model in CA Spectrum. For more information, see the *CA Spectrum Administrator Guide*.

**Notes**

(Optional) Includes any information about the SLA you want to enter not covered in the Description field.

**Expiration Date**

Specifies the date a recurring SLA period expires. Check the box, and enter the date in the field that appears. If the date falls within an SLA period, the SLA stays in effect to the end of the period.

**Period**

Specifies the interval during which the SLA is in effect. Select the period from the drop-down list, or click Create to [create an SLA period](#) (see page 111).

4. Configure one or more guarantees for the SLA.
5. Associate a service to the SLA by moving it from the Available Services list to the SLA Services list.
6. Click Create.

The SLA is created.

**More information:**

[Create a Guarantee for a Service, Sub-Service, or Resource Monitor](#) (see page 107)

[Create a Guarantee for a Top-Level Service](#) (see page 106)

[Create an SLA Template](#) (see page 114)

[Examples: Create a Guarantee for an SLA](#) (see page 135)

[Edit an SLA](#) (see page 112)

[Edit an SLA Template](#) (see page 115)

## Create an SLA From an SLA Template

An SLA template is an SLA configuration that you create and save as a template and from which you can create multiple SLAs. When you create an SLA from an SLA template the SLA inherits the template settings. You can tailor SLA settings that are inherited from a template except for the following items:

- **Guarantees** — The guarantees in an SLA template can only be modified in the SLA template workspace, and any changes to the guarantees extend to only those SLAs created from the template that have been kept in sync with the template.
- **Period** — The period in an SLA template can only be modified in the SLA template workspace, and any changes to it extend to only those SLAs created from the template that have been kept in sync with the SLA template.

Consider the following information before you create an SLA from an SLA template:

- When you create an SLA from an SLA template, you can select whether to keep the SLA in sync with (or associated with) the SLA template. If you select to keep the SLA in sync with the template, all changes to the template guarantees and SLA period extend to the SLA. If you select not to keep the SLA in sync with the template, changes to the template guarantees and the SLA period do not extend to the SLA created from the template.
- The settings for any SLA created from an SLA template that has been deleted convert to local settings.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the SLAs tab and then click Create From Template.  
The Select SLA Template dialog appears.
3. Select the SLA template that you want to use to create the SLA and click OK.  
The Create SLA From Template dialog appears.
4. Configure the settings that can be edited as required.
5. Select or clear the option to keep the SLA in sync with the template. You can also clear the option anytime after you have created the SLA from the template.
6. Click Create.  
The SLA is created from an SLA template.

**More information:**

[SLA Templates](#) (see page 114)



## Guarantee Types

You can specify one of the following guarantees for a service or a resource monitor:

### Availability

You can specify an availability threshold that is expressed as a percentage of the period for which the service is available or the amount of time in seconds that the service must not exceed for the period. This guarantee interprets the time that a service or a resource monitor service health value is down as outage time. You can also specify these supplemental thresholds when you specify an availability guarantee:

#### Mean Time Between Failure (MTBF)

$(\text{total time between failures}) / (\text{total number of failures} - 1)$

If the interval between failure falls below this value, the status of the guarantee changes to At Risk and CA Spectrum generates a Major (Orange) alarm for the SLA. Any time the interval between failures exceeds this value, CA Spectrum clears the At Risk alarm. If the guarantee is At Risk at the end of the period, CA Spectrum generates a Critical (Red) alarm for the SLA.

#### Mean Time to Repair (MTTR)

$(\text{total outage time}) / (\text{total number of outages})$

If the average outage duration exceeds this value, the status of the guarantee changes to At Risk and CA Spectrum generates a Major (Orange) alarm for the SLA. Any time the average duration falls below this value, CA Spectrum clears the "At Risk" alarm. If the guarantee is At Risk at the end of the period, CA Spectrum generates a Critical (Red) alarm for the SLA.

#### Maximum Outage Time (MOT)

If the service or resource monitor has an outage that exceeds this value, The SLA is violated, and CA Spectrum generates a Critical alarm for the SLA.

### Response Time

Lets you define a threshold for monitoring a service or resource monitor that determines its service health from the results of response time (or performance) tests. This guarantee interprets the time that a service or a resource monitor service health value is degraded as outage.

## Create a Guarantee for a Top-Level Service

You can create a guarantee for a top-level service, which lets you create a guarantee for an entire service but not to any sub-services and resource monitors.

### Follow these steps:

1. Select a threshold type: Availability or Response Time:
  - Specify a Violation Threshold value for % uptime per period or seconds of outage time per period for the threshold(s). The default uptime percentage is 99.9.
  - If you specify an availability threshold, specify values for any or all of the MTBF, MTTR, and MOT thresholds.

Service Manager provides a name for the guarantee using this format: *Threshold type - SLA name*. For example: Availability - Web Service SLA. The following is an example availability guarantee that includes an uptime threshold and MTBF, MTTR, and MOT supplemental thresholds:

The screenshot shows a 'Guarantees' configuration window. It has two main sections: 'Availability' and 'Response Time'. The 'Availability' section is selected with a checkmark. It includes a 'Violation Threshold \*' field set to '99.999' and a dropdown menu set to '% uptime per period'. Below this, there are three sub-thresholds: 'MTBF (Mean Time Between Failure) Threshold' (checked, value '5', unit 'Days'), 'MTTR (Mean Time To Repair) Threshold' (unchecked, value '0', unit 'Days'), and 'MOT (Maximum Outage Time) Threshold' (checked, value '0', unit 'Days'). The 'Response Time' section is unchecked and shows a 'Violation Threshold \*' field and a dropdown menu set to '% uptime per period'.

Guarantee is created for a top-level service.

### More information:

[Create an SLA](#) (see page 101)

[Examples: Create a Guarantee for an SLA](#) (see page 135)

## Create a Guarantee for a Service, Sub-Service, or Resource Monitor

You can create a guarantee for a service, a sub-service, or a resource monitor.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the SLAs tab, select the SLA for which you want to create the guarantee, click the Guarantees tab, and then click the Create button.

The Create Guarantee dialog appears.

3. Specify the following guarantee properties:

**Guarantee Name**

Identifies the guarantee model.

**Control**

Specifies whether the guarantee is Enabled or Disabled during the current SLA period. When you disable a guarantee, it does not accumulate outage time during the SLA period. Generally a guarantee is created in the disabled state if SLA and the services it guarantees are still be defined. However, there may be scenarios where due to alterations of a specific service it makes sense to periodically disable one of more guarantees.

**Guarantee Type**

Specifies the type of guarantee: Availability or Response Time.

**Outage Type**

Specifies whether the guarantee accumulates Down or Degraded time. response time guarantees accumulate only degraded time, while availability guarantees can be configured to accumulate either down time or degraded time.

### Accumulation Method

Service Manager provides two accumulation methods: Straight time and Per Resource.

*Straight time* means that the guarantee records outage time that corresponds exactly to service outage time. One minute of service outage time produces one minute of outage time for the guarantee. Straight time is almost always the appropriate configuration for a guarantee. Under rare circumstances per resource configurations are used to record time that is based on the resources contributing to the service outage.

*Per resource* guarantee is only used for a specific type of SLA which actually guarantees specific availability of individual service resources as opposed to the service as a whole. The result is the guarantee can record outage time in excess of the actual service outage time. Once again per resource guarantees are specialized and uncommon.

### Violation Threshold *n*% uptime per period or seconds of outage time per period

Specifies a service availability threshold that if during the current SLA period results in a critical alarm on the SLA model. The default uptime percentage is 99.9.

### (Optional) Generate warning alarm after accumulating *n*% of allowed outage time

The warning percentage is a percentage of the violation threshold in terms of allowable outage time. When a guarantee becomes warned a major alarm is generated on the SLA. This option lets you take action before the SLA becomes violated.

4. Select the Service or Resource Monitor tab, and then select a service or resource monitor by moving it from the Available Services and Groups box to the Service or Group being Measured box.
5. (Optional) [Specify “business hours” intervals](#) (see page 109) during which you want the guarantee in effect.

**Note:** By default a guarantee is always active, that means it records outage time 24x7.

6. Click Create.  
The guarantee is created.

### More information:

[Create an SLA](#) (see page 101)

[Edit a Guarantee Template](#) (see page 117)

[Examples: Create a Guarantee for an SLA](#) (see page 135)

[Create a Guarantee Template](#) (see page 116)

[Edit a Guarantee](#) (see page 110)

## Specify Business Hours for a Guarantee

If you are creating a guarantee for an SLA that measures service availability or performance for a particular portion of a day, or “business hours,” you can identify those hours in the guarantee. This means that the guarantee accumulates service outage time only during the business hours, and the SLA availability threshold applies only to the business hours.

You can also specify multiple intervals for a guarantee, as long as they do not overlap. For example, if you want a guarantee to watch a service or resource monitor from 7 AM to 5 PM on Monday, Wednesday, and Friday, 6 AM to 6 PM on Tuesday and Thursday, and continuously throughout the weekend. You can specify these three schedules in the guarantee.

For guarantees which specify a percentage of availability, it is important to understand that the availability calculation is made against the time the guarantee is active. This means that a guarantee defining business hours can have less available outage time than one with the same availability threshold that does not define business hours.

### Follow these steps:

1. [Open the Service Editor dialog](#) (see page 18).
2. Click the SLAs tab, select the SLA that contains the guarantee you want to modify, and click the Guarantees tab.
3. Select the guarantee that you want to specify business hours for and click Edit.
4. Click the Business Hours tab, and clear Always Active, as shown in the following figure:

The screenshot shows the 'Guarantee' dialog box with the 'Business Hours' tab selected. At the top, there are two tabs: 'Service or Resource Monitor' and 'Business Hours'. Below the tabs, there is a checkbox labeled 'Always Active' which is currently unchecked. Under the heading 'Current Schedules:', there is an empty list box. To the right of this list are two arrow buttons (right and left). Below the list box is a 'Filter:' text box and a 'Displaying 0 of 0' status indicator, followed by a 'Create...' button. On the right side of the dialog, under the heading 'Available Schedules:', there is a list box containing four schedule entries: 'Every day from 6:00 PM - 7:00 AM', 'Every day from 7:00 AM - 6:00 PM', 'Every week Mon thru Fri from 6:00 AM - 11:00 PM', and 'Every week Mon thru Fri from 8:00 AM - 8:00 PM'. Below this list box is a 'Filter:' text box and a 'Displaying 6 of 6' status indicator.

**Note:** If you want the guarantee in effect continuously throughout the SLA period, click the Business Hours tab and select Always Active (default).

5. Select and move one or more intervals from the Available Schedules list to the Current Schedules list.
6. (Optional) If you require a custom interval, take the following steps:
  - a. Click Create.
  - b. Configure the interval and click OK.  
The custom interval is added to the Available Schedules list.
  - c. Select and move the custom interval from the Available Schedules list to the Current Schedules list.
7. Click OK.  
Business hours are specified for the guarantee.

**More information:**

[Create a Guarantee for a Service, Sub-Service, or Resource Monitor](#) (see page 107)

## Edit a Guarantee

You can edit a guarantee anytime after you create it; however changes to thresholds or business hours are not recommended.

Consider the following points before editing a guarantee:

- Modified thresholds take effect during the current period, this may not be desirable.
- A guarantee begins a new down or degraded time tally when you change the service or resource monitor with which it is associated during the current period.
- A guarantee modifies its outage time tally during a period if an outage has occurred and ended during the period and the outage has had its status modified (to exempt for example) during the period.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the SLAs tab, select an SLA from the list of SLAs, click the Guarantees tab (in the lower panel) and click Edit.  
The Edit Guarantee dialog appears.
3. Edit the settings, as described in [Create a Guarantee for a Service, Sub-Service, or Resource Monitor](#) (see page 107) and click OK.  
The guarantee is edited.

## Delete a Guarantee

You can delete a guarantee for an SLA as your requirements for the SLA change.

**Important!** Delete with caution. Deleting the only guarantee for an SLA renders it inoperative.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the SLAs tab, and then select the SLA from which you want to delete a guarantee.
3. Click the Guarantees tab, select the guarantee that you want to delete, and then click Delete (in the lower panel).
4. Respond to the confirmation message that appears to complete the deletion.

The guarantee is deleted.

## Create an SLA Period

Service Manager provides two default SLA periods that commence at the following times for the time zone where the SpectroSERVER is located:

- On the first of every month at 12:00 AM
- On the 15th of every month at 12:00 AM

You can specify one of these periods when you create or edit an SLA, or you can create and specify a custom period that meets your particular service contract requirements. Service Manager saves the periods that you create so you can use them for other SLAs.

**Follow these steps:**

1. Click the Create button next to the Period field in the Create SLA dialog or the Edit SLA dialog.

The Create Period dialog appears.

2. Specify a period, and optionally a description, and then click OK. Repeat as necessary to create and save more periods.

The SLA period is created.

**More information:**

[Create an SLA](#) (see page 101)

## Edit an SLA

Periodically it is necessary to alter the configuration of a SLA model. Service Manager allows certain edits, but you can take care in the types of changes that are made. If the required change involves alteration of guarantee thresholds, business hours or the SLA period it is recommended that you do not make these edits to an active SLA. The period and guarantee thresholds are the essence of the SLA. If the guarantees stipulations are changed, it implies that a new SLA is available.

Therefore it is important that the real-time status of an SLA, and its historical reported status are based on complete periods with consistent thresholds. Consider how confusing data may look if an SLA recorded that same amount of outage time across two periods, but is compliant for one period and violated for the next due to a threshold change.

It is recommended to set an expiration date of the SLA to correspond with the end of the current period, instead editing a standing SLA. Later create a SLA with the new threshold settings, and set its control value to Inactive Until Next Period. This confirms that the current SLA completes the period with its existing configuration and the new SLA takes over seamlessly at the beginning of the next period.

If you change the service that is associated with the SLA, you can see the following output:

- Service Manager resets the start time of the SLA.
- All of the SLA guarantees go to the Initial (Blue) state, which means that you must associate a new service or resource monitor to each guarantee.

### Follow these steps:

1. [Open the Service Editor](#) (see page 18).
2. Click the SLA tab, select the SLA you want to edit from the list, and then click Edit.

The list field, Template Name, indicates whether the SLA was created from an SLA template. If it was, the field lists the source SLA template name.

The Edit SLA dialog appears. If you select an SLA created from an SLA template, the dialog includes the Keep in sync with the template selection box. You can clear the sync option if you want to disassociate the SLA from its source SLA template and edit all SLA fields. Otherwise, you can edit only those settings that are not managed in the source SLA template.

3. Edit the settings, as described in [Create an SLA](#) (see page 101) and click OK.

The SLA is edited.



## Delete an SLA

You can delete an SLA anytime after you create it. However, you would typically not delete an SLA that is actively monitoring a service.

**Important!** Be cautious when deleting SLAs because you may inadvertently delete an SLA that is actively monitoring a service.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the SLAs tab, select the SLA you want to remove from CA Spectrum, and then click Delete.
3. Respond to the confirmation message that appears to complete the deletion.  
The SLA is deleted.

## Associate a Customer with an SLA

Service Manager lets you associate one or more customers with an SLA to help you track and manage SLAs and customers. The status of the customer model is not impacted by the status of the SLAs associated to it. You can generate SLA reports with CA Spectrum Report Manager based on SLA-customer associations.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the SLAs tab, select the SLA you want to associate with a customer, click the Customers tab, and then click Select SLA Customers.  
The Select Customers dialog appears.
3. Move the customer that you want to associate with the SLA from the Available Customers list to the Customers that use this SLA list and click OK.  
The customer is associated with an SLA.

## SLA Templates

An SLA template includes configuration settings that are inherited by the SLAs that you can create from the template. It lets you create multiple SLAs with similar settings for different customers and services without having to configure each SLA individually. The use of SLA templates is common in traditional service provider environment that offers similar SLAs to multiple customers. The SLA template also allows you to make changes or additions to all associated SLAs by editing the template itself. Although this can be a convenient feature, be cautious when making edits guarantee thresholds or business hours setting.

The following SLA template settings cannot be edited in an SLA that has been created from a template while the SLA is in sync with the template:

- Period
- Guarantees

All other inherited settings can be modified in the SLA. All settings can be modified, however, if an SLA is disassociated (in sync option deselected) from its parent template.

The guarantees that you create for SLA templates are referred to as guarantee templates. You create and manage guarantee templates in the SLA Template workspace the same way you do with guarantees in the SLA workspace.

### More information:

[Create an SLA From an SLA Template](#) (see page 103)

## Create an SLA Template

You can create as many SLA templates as you require, but the templates must have unique names.

### Follow these steps:

1. [Open the Service Editor](#) (see page 18).
2. Click the SLA Templates tab and click Create.

The Create SLA Template dialog appears.

3. [Specify settings for the SLA template](#) (see page 101).

**Note:** SLA Templates are often created with a control of Inactive. It lets you instantiate the actual SLAs in the inactive state, and activate them when appropriate.

4. (Optional) Configure basic guarantee settings for the template. You can specify more detailed guarantee settings or a new guarantee for the template (such as guarantee template) after you create the template.
5. Click Create.

The SLA template is created.

## Edit an SLA Template

You can edit an SLA template anytime before or after it is in effect.

Consider the following information before editing an SLA template:

- Changes to the period and guarantees in a template extend to all SLAs created from it that are in sync with the template. This may not be desirable.
- When you delete a template, the associations between the SLAs created from it that are in sync with the template are severed. This means you can edit all period and guarantee settings in the SLAs that were once managed exclusively in the former template.

### Follow these steps:

1. [Open the Service Editor](#) (see page 18).
2. Click the SLA Templates tab, select the template that you want to edit from the list, and click Edit.

The Edit SLA Template dialog appears.

3. Edit the settings, as described in [Create an SLA](#) (see page 101), and click OK.

The SLA template is edited.

## Delete an SLA Template

You can delete an SLA template anytime after you create it. When you delete an SLA template, all SLAs created from the template that are kept in sync with it become fully editable.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the SLA Templates tab, select the template from the list you want to remove from CA Spectrum, and click Delete.
3. Respond to the confirmation message that appears to complete the deletion.

The SLA template is deleted.

## Guarantee Templates

You can create guarantee templates for SLA templates. You can edit guarantee template settings, and those changes extend to all SLAs created from the SLA template that includes the guarantee template.

You can modify guarantee templates only from the SLA Templates workspace. You cannot access guarantee templates from the Guarantees tab in the SLA workspace.

You cannot specify watched services or Resources Monitors for guarantee templates. You can associate the guarantee to services and resource monitors you specify when you create an SLA from an SLA template that includes the guarantee template.

**Note:** Be cautious when editing guarantee templates for the reasons that are outlined in the Edit an SLA section.

## Create a Guarantee Template

Guarantee templates can be created in the Create SLA Template dialog or in the Create Guarantee Template dialog. This section describes the latter method.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the SLA Templates tab, select the SLA template for which you want to create the guarantee template, click the Guarantee Template tab, and then click Create.  
The Create Guarantee Template dialog appears.
3. [Configure settings](#) (see page 107), and then click Create.

The guarantee template appears under the Guarantee Template tab for the SLA template and is inherited by all SLAs created from the template.

## Edit a Guarantee Template

You can edit a guarantee template any time after you create it. Because they are changed within the context of an SLA template and all SLAs created from it that are in sync with the template are also changed. Be cautious when editing guarantee templates for the reasons that are outlined in the Edit an SLA section.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the SLA Templates tab and select the SLA Template that includes the guarantee template you want to edit from the list of templates.
3. Click the Guarantee Templates tab to display guarantee templates for the selected SLA template.
4. Select the guarantee template that you want to edit from the list, and then click Edit (in the lower panel).

The Edit Guarantee Template dialog appears.

5. Edit the settings as necessary as described in Edit a Guarantee, and click OK. For more information, see [Create a Guarantee for a Service, Sub-Service, or Resource Monitor](#) (see page 107).

## Delete a Guarantee Template

You can delete a guarantee template as your requirements change.

**Important!** Delete with caution. Deleting the only guarantee template for an SLA template renders the SLAs created from and in sync with the template inoperative.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Click the SLA Templates tab and select the SLA template from the list of templates from which you want to delete a guarantee template.
3. Click the Guarantee Templates tab, select the guarantee template that you want to delete, and then click Delete (in the lower panel).
4. Respond to the confirmation message that appears to complete the deletion.

The guarantee template is deleted.

# Chapter 6: Creating Service Management Components with Modeling Gateway

---

You can create service management component models using the CA Spectrum Modeling Gateway Toolkit to define service component model configurations in XML input files and import the files into CA Spectrum. Defining service management models with the Modeling Gateway Toolkit and importing them into CA Spectrum instead of creating them with Service Editor is advantageous in the following ways:

- Modeling Gateway lets you define new models in bulk. Once you have created a particular service management model, you can use the XML input file for the model as a template for creating other models of that type.
- You can edit service management models that are imported through Modeling Gateway either by using the Service Editor or by simply editing and re-importing the original XML file.
- There can be an opportunity to automate the creation of service management models by producing an xml file for import that is based on an external data source. You can verify the Modeling Gateway capabilities and prerequisites. For more information, see the *Modeling Gateway Toolkit Guide*.

This section contains the following topics:

[About the XML Framework](#) (see page 120)

[Service Models](#) (see page 121)

[Policies and Watched Attributes](#) (see page 122)

[Example: Services That Monitor Resources Directly](#) (see page 123)

[Example: Services That Monitor Resources in Resource Monitors](#) (see page 124)

[Example: Using XML to Define a Service Template](#) (see page 126)

[Example: Define a Service Maintenance Schedule](#) (see page 133)

[Example: Define an Alarm Exemption List for a Service or Resource Monitor](#) (see page 133)

[Example: Associate an SLA to a Service](#) (see page 135)

[Examples: Create a Guarantee for an SLA](#) (see page 135)

[Example: Define an SLA](#) (see page 136)

[Example: Define a Customer and a Customer Group](#) (see page 137)

[Example: Import XML Input Files](#) (see page 140)

[Service Attributes \(SM\\_Service\)](#) (see page 141)

[Monitor Resource Monitor Attributes \(SM\\_AttrMonitor\)](#) (see page 142)

[Customer Group Attributes \(SM\\_CustomerGroup\)](#) (see page 143)

[Customer Attributes \(SM\\_Customer\)](#) (see page 143)

[SLA Attributes \(SM\\_SLA\)](#) (see page 144)

[Guarantee Attributes \(SM\\_Guarantee\)](#) (see page 145)

[Schedule Attributes \(Schedule\)](#) (see page 147)

## About the XML Framework

The hierarchy models (SM\_Service\_Mgt, SM\_ServiceMgr, SM\_SLA\_Mgr, CustomerManager) must be arranged and named in the XML input file according to the following example.

The following example illustrates the basic framework of the XML input file. Each service management component is added within the appropriate section of the file. Service models can be created within the SM\_ServiceMgr block, or within other services. Customer models and Customer Group models are created within the CustomerManager block. Finally the SLA models are created within the SLA\_Mgr block.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE Import SYSTEM ".import.dtd">

<Import>

<SM_Service_Mgt
  name="Service Management"
  containment_relation="SImHasServiceComponent">

  <SM_ServiceMgr
    name="Services"
    containment_relation="SImContains">

  </SM_ServiceMgr>

  <SM_SLA_Mgr
    name="SLAs"
    containment_relation="SImContainsSLAs">

  </SM_SLA_Mgr>

  <CustomerManager
    name="Customers"
    containment_relation="Groups_Customers">

  </CustomerManager>

</SM_Service_Mgt>

</Import>
```

The following image displays the hierarchy that is represented in the OneClick Console.





**Important!** Although the name and containment\_relation attribute values in the framework tags must be adhered to in the XML file you import, the individual service, resource monitor, customer, SLA, and guarantee models you import must have unique names. This differs from the OneClick client. Because modeling gateway uses name for uniqueness. Services with the same name are interpreted as the same model.

## Service Models

Service models must be defined within the SM\_ServiceMgr tag in the XML file that you import. For each landscape where Service Manager is installed, there is only one SM\_ServiceMgr model with the model name, Services.

Each service model in a given landscape must be either SImContains by the ServiceManager or SImMonitors by another service model.

To use Modeling Gateway effectively, understand the mapping between policies and policy IDs. For example, if a service monitors other services or resource monitors, it can be configured with a service health policy (IDs 6-9). Also, you can determine a policy ID for a user-created policy by setting up the Policies table in the Service Policy Editor to display IDs.

### More information:

[Monitor Resource Monitor Attributes \(SM\\_AttrMonitor\)](#) (see page 142)

[SLA Attributes \(SM\\_SLA\)](#) (see page 144)

[Guarantee Attributes \(SM\\_Guarantee\)](#) (see page 145)

[Customer Group Attributes \(SM\\_CustomerGroup\)](#) (see page 143)

[Customer Attributes \(SM\\_Customer\)](#) (see page 143)

[Service Manager Policy Descriptions](#) (see page 183)

[Customize a Service Policy Editor Information Table](#) (see page 198)

[Service Attributes \(SM\\_Service\)](#) (see page 141)

[Schedule Attributes \(Schedule\)](#) (see page 147)

## Policies and Watched Attributes

When using Modeling Gateway to model services, the value for the MonitorPolicy\_ID can be set to the id number associated with the policy the service uses. Service policy ids can be seen in the service policy editor by adding the Service Policy ID column to the service policies table. Out-of-the-box policies start at 1, user created policies start at 1000.

In addition to viewing policy ID in the service policy editor, you can see the policy id that is displayed at the following link.

`http://<server>/spectrum/slm/ policyrep.jsp`

Remember that when specifying the Monitor Policy\_ID, verify that the AttrToWatch matches the attribute for the policy that you have selected.

**Note:** If your XML file creates a mismatch between a monitor policy ID and watched attribute (for example, watching Contact Status using policy ID 2), Service Manager puts the service in a defunct condition, which is reported as an outage and can be viewed in Service Dashboard. When a service becomes default an alarm is generated on the Service Management model. This alarm is major for services which are associated to an SLA, and minor for services which are not.

### More information:

[Policy ID Mappings](#) (see page 183)

## Example: Services That Monitor Resources Directly

The following XML document configures a service named “Test Service.” It monitors the contact status of two Cisco routers and generates a critical alarm if contact to either router is lost:

```
<!-- Each SpectroSERVER will have only one -->
<!-- SM_ServiceMgr model, named "Services". -->

<SM_ServiceMgr
  name="Services"
  containment_relation="SImContains">← [This relation associates a service with resources.]

  <SM_Service
    containment_relation="SImMonitors"
    name="Test Service"
    AttrToWatch="Contact_Status"
    MonitorPolicy_ID="11"← [Enter either a SPECTRUM-provided Policy ID (1-21)
    Criticality="10"← [or ID (1000, 1001, . . .) for a policy created
    Generate_Service_Alarms="true">← [in Service Editor.]
    <Device ip_dnsname="10.253.9.7" />← [Enter device (resource) IP address or DNS name.]
    <Device ip_dnsname="10.253.9.8" />
  </SM_Service>
</SM_ServiceMgr>
```

- The Services model contains (has an SImContains relationship with) Test Service. This can make the Test service a direct child of the service manager. The Test service appears under the Services icon in the OneClick navigation panel.
- Test Service monitors (has an SImMonitors relationship with) the 10.253.9.7 and 10.253.9.8 devices and watches their Contact\_Status attributes using the Contact Status High Sensitivity policy.
- The value of Generate\_Service\_Alarms is true, indicating that when the service health value is down, degraded, or slightly degraded, CA Spectrum generates an alarm for the service.

### More information:

[Example: Import XML Input Files](#) (see page 140)

## Example: Services That Monitor Resources in Resource Monitors

The following XML document defines a service, XYZ Service, that monitors two other services (Core Routers and DNS) directly. In addition, the XYZ service defines three resource monitors by specifying SM\_AttrMonitor elements. The first resource monitor XYZ Condition monitors the contents of the XYZ Network container. The second resource monitor XYZ Response Time monitors an SPM test call XYZ\_RTM\_1. Finally, the service also defines a resource monitor, XYZ Port Status which monitors an interface model.

```
<SM_Service
    containment_relation="SlmMonitors"
    name="XYZ Service"
    Criticality="25"
    AttrToWatch="Service_Health"
    MonitorPolicy_ID="8"
    Generate_Service_Alarms="true">

    <SM_AttrMonitor
        containment_relation="SlmWatchesContainer"
        name="XYZ Condition"
        AttrToWatch="Condition"
        MonitorPolicy_ID="2">

        <Topology_Container model_type="Network" name="XYZ
Network Servers" />
    </SM_AttrMonitor>

    <SM_AttrMonitor
        containment_relation="SlmMonitors"
        name="XYZ Response Time"
        AttrToWatch="LatestErrorStatus"
        MonitorPolicy_ID="18">

        <RTM_Test name="XYZ_RTM_1" />
    </SM_AttrMonitor>

    <SM_AttrMonitor
        containment_relation="SlmMonitors"
        name="XYZ Port Status"
        AttrToWatch="Port_Status"
        MonitorPolicy_ID="15">

        <Port ip_dnsname="10.253.50.5"
        identifier_name="ifIndex"
        identifier_value="45" />
    </SM_AttrMonitor>
```

```
<SM_Service name="Core Routers"/>
<SM_Service name="DNS"/>

</SM_Service>
```

## Example: Using XML to Define a Service Template

If you encounter a scenario where many services share a common pattern or structure. You can define that structure in xml, and can use it as a common template. For example, you can build services to monitor a set of applications which are all different, but have common service modeling components. You can define the structure and import as many service models as you need from it. Some of the data can be added for import, or you can create empty services and resource monitors. You can add resources to them using the OneClick client.

The following syntax shows an example of the xml for a small service hierarchy that define a set of reusable service and resource definitions. The TMPL text represents wildcard test that can be changed to a more meaningful name for each set of services to be imported.

For this example, you can see a service which includes monitoring capability for some application servers, and associated database servers. As well as some additional some response time and performance monitoring. This example is not intended to match any specific requirements, but serves as an example to create an xml template for a set of services with common requirements.

```
<SM_Service
  containment_relation="SlmMonitors"
  name="TMPL Application Service"
  Criticality="10"
  AttrToWatch="Service_Health"
  MonitorPolicy_ID="7"
  Generate_Service_Alarms="true">

  <SM_Service
    containment_relation="SlmMonitors"
    name="TMPL Application Servers"
    Criticality="10"
    AttrToWatch="Service Health"
    MonitorPolicy_ID="9"
    Generate_Service_Alarms="true">

    <SM_Service
      containment_relation="SlmMonitors"
      name="TMPL Application Server 1"
      Criticality="10"
      AttrToWatch="Service Health"
      MonitorPolicy_ID="7"
      Generate_Service_Alarms="true">

      <SM_AttrMonitor
        containment_relation="SlmMonitors"
        name="TMPL App Host 1"
        AttrToWatch="Condition"
        MonitorPolicy_ID="3"
```

```
Cause_List_Control="2"
Special_Cause_List="0x1106f-0x11232">

    // Excludes all eHealth alerts

</SM_AttrMonitor>

<SM_AttrMonitor
    containment_relation="SlmMonitors"
    name="TMPL App Server 1 Critical Processes"
    AttrToWatch="Condition"
    MonitorPolicy_ID="3">

</SM_AttrMonitor>

<SM_AttrMonitor
    containment_relation="SlmMonitors"
    name="TMPL App Server 1 System Resources"
    AttrToWatch="Condition"
    MonitorPolicy_ID="3">

</SM_AttrMonitor>

<SM_AttrMonitor
    containment_relation="SlmMonitors"
    name="TMPL App Server 1 Connection"
    AttrToWatch="Response Time"
    MonitorPolicy_ID="19">

</SM_AttrMonitor>

<SM_AttrMonitor
    containment_relation="SlmMonitors"
    name="TMPL App Server 1 Performance"
    AttrToWatch="Condition"
    MonitorPolicy_ID="3"
    Cause_List_Control="1"
    Special_Cause_List="0x1120a,0x11219">

    // Includes 2 specific eHealth alerts only
</SM_AttrMonitor>

</SM_Service>

</SM_Service>

<SM_Service
    containment_relation="SlmMonitors"
    name="TMPL Database Servers"
```

```
Criticality="10"
AttrToWatch="Service Health"
MonitorPolicy_ID="6"
Generate_Service_Alarms="true">

<SM_Service
  containment_relation="SlmMonitors"
  name="TMPL Database Server 1"
  Criticality="10"
  AttrToWatch="Service Health"
  MonitorPolicy_ID="7"
  Generate_Service_Alarms="true">

  <SM_AttrMonitor
    containment_relation="SlmMonitors"
    name="TMPL DB Host 1"
    AttrToWatch="Condition"
    MonitorPolicy_ID="3"
    Cause_List_Control="2"
    Special_Cause_List="0x1106f-0x11232">

    // Excludes all eHealth alerts

  </SM_AttrMonitor>

  <SM_AttrMonitor
    containment_relation="SlmMonitors"
    name="TMPL DB Server 1 Critical Processes"
    AttrToWatch="Condition"
    MonitorPolicy_ID="3">

  </SM_AttrMonitor>

  <SM_AttrMonitor
    containment_relation="SlmMonitors"
    name="TMPL DB Server 1 System Resources"
    AttrToWatch="Condition"
    MonitorPolicy_ID="3">

  </SM_AttrMonitor>

  <SM_AttrMonitor
    containment_relation="SlmMonitors"
    name="TMPL DB Server 1 Connection"
    AttrToWatch="Response Time"
    MonitorPolicy_ID="19">

  </SM_AttrMonitor>
```



```
        </SM_Service>

    </SM_Service>

    <SM_Service
        containment_relation="SlmMonitors"
        name="TMPL Application Performance & Response Time"
        Criticality="10"
        AttrToWatch="Service Health"
        MonitorPolicy_ID="7"
        Generate_Service_Alarms="true">

        <SM_Service
            containment_relation="SlmMonitors"
            name="TMPL Application Response Time"
            Criticality="10"
            AttrToWatch="Response Time"
            MonitorPolicy_ID="20"
            Generate_Service_Alarms="true">

            <SM_Service>

            <SM_Service
                containment_relation="SlmMonitors"
                name="TMPL Application Performance
                Criticality="10"
                AttrToWatch="Condition"
                MonitorPolicy_ID="4"
                Cause_List_Control="1"
                Special_Cause_List="0x1106f-0x11232">

                    // Includes eHealth alerts only

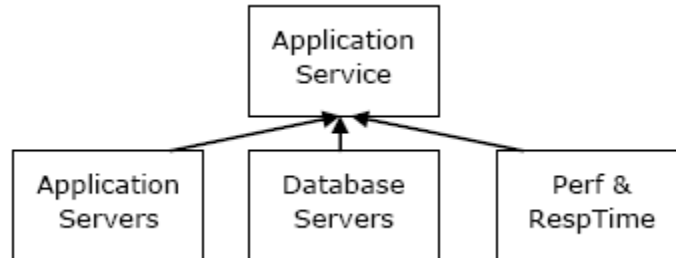
                Generate_Service_Alarms="true">

            </SM_Service>

        </SM_Service>

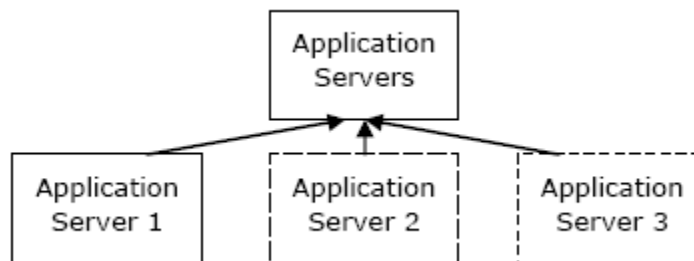
    </SM_Service>
```

Now let us review each component in the xml to understand its purpose and how it fits within the service hierarchy that is defined by the xml. At the top of the hierarchy we have the application Service which has three direct child services:



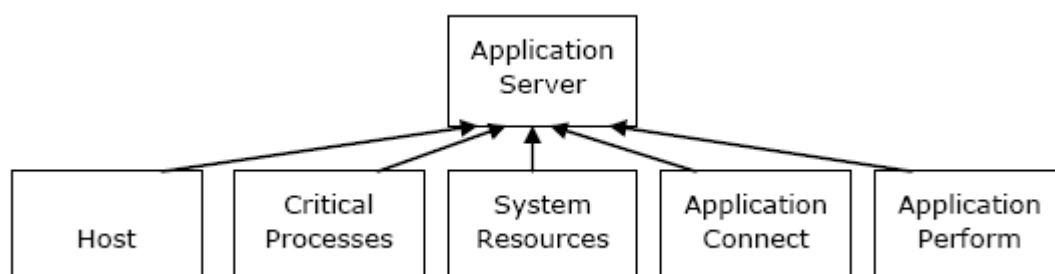
The Application Service monitors each of its child service with a high sensitivity policy. Therefore, the Application Service can have a service health equal to that of its worst direct resource.

The Application Servers service is designed to monitor child services which represent individual servers. Each server is represented by a service model with five resource monitors. The xml example contains only Application Server 1, but you could add this section for as many servers as necessary.

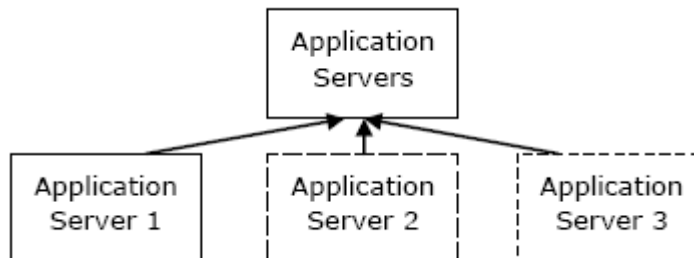


The Application Servers service could monitor the health of individual child services with a redundancy, percentage of low sensitivity policy.

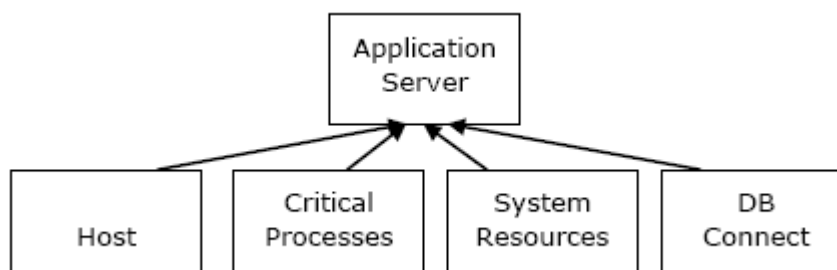
Each Application Server service; however, would monitor its resource monitors with a high sensitivity policy. The resource monitors focus on the host itself, critical processes, system resources, application connection, and application performance. You can notice that the Host resource monitor excludes CA eHealth notifications which are performance-based. The Application Performance resource monitor is only impacted by CA eHealth notifications. Both resource monitors would likely have the same host model as a resource, but are affected by different types of resource outages. This allows you to determine if the service outage is related to availability or performance.



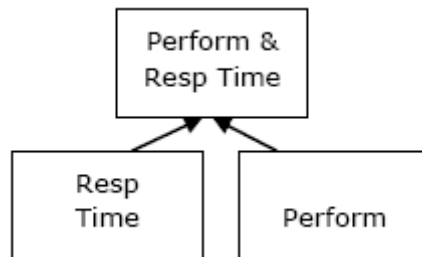
The Database Servers service has a similar structure to Application Servers. Multiple individual servers can be supported.



Each individual server supports four resource monitors. These detect faults on the host model, critical database processes, system resources, and database connections. Once again, you can see that the Host resource monitor excludes CA eHealth notifications.



Finally the service also includes a component for monitoring performance and response time. Application Response Time sub service would focus on monitoring SPM tests in CA Spectrum. You can use default response time policies, or perhaps develop a custom policy for average response time. The Performance sub service would detect resource faults that are based on CA eHealth performance notifications that are sent to CA Spectrum. The performance service would likely use a set of host models as its resources where CA eHealth notifications are mapped to resource alarms.



This example is not designed to fit a specific scenario, but provide you with an example of how modeling gateway can be used to model common patterns in your environment.

If you are able to identify all of the resources, they can be added to the service and resource monitor elements in the xml file. Even if you are not sure of all service resources, services and resource monitors can still be imported and empty of resources. The empty services and resource monitors appear with blue icons in CA Spectrum prompting you to the need to fill in the resources.

## Example: Define a Service Maintenance Schedule

In addition to defining the structure of a service and its resources you can also specify a maintenance schedule for service. The following XML segment defines a maintenance schedule for a service named "ABC Service", using an existing schedule model.

```
<SM_Service
  containment_relation="MaintenanceScheduledBy"
  name="ABC Service">

  <Schedule name="Every day from 6 PM - 7 AM"
    SCHED_Recurrence="2"
    SCHED_Duration="46800"
    SCHED_Start_Hour="18"
    SCHED_Start_DoM="0"
    SCHED_DayBitMask="0"
    SCHED_Start_Day="0"
    SCHED_Description=""
    SCHED_Start_Year="0"
    SCHED_Start_DoW="0"
    SCHED_Start_MoY="0"
    SCHED_Start_Minute="0"
    SCHED_Start_Month="0"
    SCHED_Daily_Repeat_Limit="2"
    SCHED_Recurrence_Multiplier="1"/>

</SM_Service>
```

### More information:

[Service Maintenance Schedule Management](#) (see page 63)

## Example: Define an Alarm Exemption List for a Service or Resource Monitor

You can specify an alarm type exclusion list for a service using modeling gateway. This setting applies to the service model, and can be used in lieu of any setting that is made at the policy level. This xml configuration is equivalent to specifying the alarm type exemption in the Exemptions tab of the Service Editor. If the configuration you want to specify for this service is defined within a policy, specify the policy ID.

The following XML segment specifies the three alarms (0xabcd0001, 0xabcd0001, 0xabcd0002) that are the only alarm types to affect (Cause\_List\_Control="1") the service:

```
<SM_Service
  containment_relation="SlmMonitors"
  name="Access Routers"
  Criticality="30"
  AttrToWatch="Condition"
  Cause_List_Control="1"
  Special_Cause_List="0xabcd0001,0xabcd0001,0xabcd0002"
  MonitorPolicy_ID="2"
  Generate_Service_Alarms="true">

  <Device ip_dnsname="10.253.9.16" />
  <Device ip_dnsname="10.253.9.17" />
  <Device ip_dnsname="192.168.152.5" />
  <Device ip_dnsname="172.19.17.174" />
</SM_Service>
```

The following XML document specifies a range of alarms (0xeeee0000-0xeeee002b) that can be excluded from impacting the health of the service (Cause\_List\_Control="2") the resource monitor:

```
<SM_AttrMonitor
  containment_relation="SlmMonitors"
  name="Access Routers"
  Criticality="30"
  AttrToWatch="Condition"
  Cause_List_Control="2"
  Special_Cause_List="0xeeee0000-0xeeee002b"
  MonitorPolicy_ID="2">

  <Device ip_dnsname="10.253.9.16" />
  <Device ip_dnsname="10.253.9.17" />
  <Device ip_dnsname="192.168.152.5" />
  <Device ip_dnsname="172.19.17.174" />
</SM_AttrMonitor>
```

**More information:**

[Specify the Alarm Types That Affect or Do Not Affect Service Health](#) (see page 57)

## Example: Associate an SLA to a Service

Service models can be associated to SLA models using modeling gateway. This association does not create any specific guarantee models, or associate the service to any existing guarantee model. Associations for guarantees must be done explicitly and can be covered later in this document.

The following sample XML shows how to associate an SLA to a service:

```
<SM_SLA
  containment_relation="SImGuarantees"
  name="Acme Service Level Agreement">

  <SM_Service name="Acme"/>
</SM_SLA>
```

## Examples: Create a Guarantee for an SLA

Guarantee models are created within an SLA element, and can be associated to a service or resource monitor model. The following XML shows how to create a guarantee for an SLA call Acme Service Level Agreement. You can call the guarantee as Engineering Guarantee and record outage time when the Engineering service is Down.

```
<SM_SLA
  containment_relation="SImHasGuarantee"
  name="Acme Service Level Agreement">

  <SM_Guarantee
    containment_relation="SImIsMeasuredBy"
    name="Engineering Guarantee"
    GuaranteeControl="1"
    GuaranteeType="0"
    ServiceHealthType="1"
    WarningThresholdPercent="80.5"
    ViolationThresholdPercent="99.5"
    GuaranteeNotes="Tracks Down Time For Engineering Service"
    GuaranteeDescription="Availability Guarantee for Acme Engineering"
    MOT_Threshold="300"
    MTBF_Threshold="300"
    MTTR_Threshold="300">

    <SM_Service name="Engineering"/>

  </SM_Guarantee>
</SM_SLA>
```

Guarantee business hours can be specified using xml, by defining the schedule. The following example shows how a schedule named Business Hours can be associated to the Engineering Guarantee model.

```
<SM_Guarantee
  containment_relation="SlmSchedulesGuarantee"
  name="Engineering Guarantee"
  GuaranteeType="0">

  <Schedule
    name="Business Hours"
    SCHED_Recurrence="2"
    SCHED_Daily_Repeat_Limit="0"
    SCHED_Duration="25200"
    SCHED_Recurrence_Multiplier="1"
    SCHED_Start_DoM="0"
    SCHED_Start_DoW="0"
    SCHED_Start_Hour="8"
    SCHED_Start_Minute="0"
    SCHED_Start_Month="0"
    SCHED_Start_Day="0"
    SCHED_DayBitMask="0"
    SCHED_Start_Year="0"
    SCHED_Start_MoY="0"
    SCHED_Description="Standard Business Hours 8AM Start"/>
  </SM_Guarantee>
```

### More information:

[Create an SLA](#) (see page 101)

[Create a Guarantee for a Service, Sub-Service, or Resource Monitor](#) (see page 107)

[Create a Guarantee for a Top-Level Service](#) (see page 106)

[SLA Attributes \(SM SLA\)](#) (see page 144)

[Guarantee Attributes \(SM Guarantee\)](#) (see page 145)

## Example: Define an SLA

The SLA Manager is the top-level model for all SLAs. Each SpectroSERVER has one SLA Manager model.



In addition to creating the SLA and its guarantees, the SLA period can be defined using modeling gateway. The following XML example shows how to define an SLA period, by specifying the period schedule.

```
<SM_SLA_Mgr
  name="SLAs"
  containment_relation="SImContainsSLAs"
>
<SM_SLA
  containment_relation="SlaPeriod"
  name="Acme Service Level Agreement"
  SLA_Control="1"
  SLA_ExpirationDate="1514696400"
  SLA_Notes="Manages SLA for Acme Services"
  SLA_Description="Acme Management Technologies Internal Service Level
Agreement">

  <Schedule
    name="Daily SLA Schedule"
    SCHED_Recurrence="2"
    SCHED_Duration="0"
    SCHED_Start_Hour="0"
    SCHED_Start_DoM="0"
    SCHED_DayBitMask="0"
    SCHED_Start_Day="0"
    SCHED_Description=""
    SCHED_Start_Year="0"
    SCHED_Start_Dow="0"
    SCHED_Start_MoY="0"
    SCHED_Start_Minute="0"
    SCHED_Start_Month="0"
    SCHED_Daily_Repeat_Limit="0"
    SCHED_Recurrence_Mutiplier="1"/>

</SM_SLA>

</SM_SLA_Mgr>
```

**More information:**

[SLA Attributes \(SM\\_SLA\)](#) (see page 144)

## Example: Define a Customer and a Customer Group

Customer models must be defined within the CustomerManager tag in the XML file. Each SpectroSERVER has one Customer Manager model, and it must be named as Customers.

The following example XML document defines a customer that is named Product Development within a customer group named XYZ Group:

```

<CustomerManager
  name="Customers"
  containment_relation="Groups_Customers">

  <SM_CustomerGroup
    name="XYZ Group" ← Specify Customer Group name here.
    containment_relation="Groups_Customers">

    <!-- This code defines a Customer and associates -->
    <!-- it with a Service. A Customer contains -->
    <!-- primary and secondary contact information, -->
    <!-- and a criticality that can effect the -->
    <!-- severity of a Service outage for Service -->
    <!-- models used by the Customer. -->

    <SM_Customer
      containment_relation="SlmUses"
      name="Product Development" ← Specify Customer name here.
      CustomerField4="Pease International TradePort"
      CustomerField5="123 Big Dr."
      CustomerField6="Portsmouth, NH 03801"
      CustomerField7="USA"
      CustomerID="11DU-156"
      Criticality="10"
      Contact_Name="Fred Flintstone"
      Contact_Title="Product Development Manager"
      Email_Address="fred@proddev.com"
      Phone_Number="123-456-7890"
      Mobile_Phone_Number="123-456-1111"
      Secondary_Contact_Name="Barney Rubble"
      Secondary_Contact_Title="Product Development Manager"
      Secondary_Phone_Number="123-456-9999"
      Secondary_Mobile_Phone_Number="123-456-8888"
      Secondary_Email_Address="barney@proddev.com">

      <SM_Service name="Development"/> ← Specify service name here.

    </SM_Customer>

    <!-- This code associates a Customer with an SLA. -->
    <SM_Customer
      containment_relation="SlmAgrreesTo"
      name="Product Development">

      <SM_SLA name="XYZ Service Level Agreement"/>

    </SM_Customer>

  </SM_CustomerGroup>

</CustomerManager>

```

**More information:**

[Create a Customer Group](#) (see page 93)

[Create a Customer](#) (see page 92)

[Customer Group Attributes \(SM CustomerGroup\)](#) (see page 143)

[Customer Attributes \(SM Customer\)](#) (see page 143)

## Example: Import XML Input Files

You can use Modeling Gateway to import your Service Manager configuration files. Use the Modeling Gateway to test the XML document in [Example: Services That Monitor Resources Directly](#) (see page 123).

**Follow these steps:**

1. Create a file containing the [example XML document](#) (see page 123) and save it to the `<$SPECROOT>/SS-Tools` directory with the file name, `slm_test1.xml`.
2. Import the file using the following command from the `<$SPECROOT>/SS-Tools` directory:

```
/topimport -vnm <vnm_name> -i slm_test1.xml -debug
```

***vnm\_name***

Is the name of the system on which the SpectroSERVER is running.

***-debug***

Generates a TIDebug.txt file, which is useful for debugging import errors.

3. If no errors are reported, device and container models are created. The output can be similar to the following output:

```
Import session started Fri, December 29, 2006, at 02:53:32 EST
Start parsing file slm_test1.xml
Start importing file slm_test1.xml
Container models created: 1
Identifying ports...
Import session finished Fri, December 29, 2006, at 02:53:38 EST
```

## Service Attributes (SM\_Service)

When creating service management models with Modeling Gateway, the XML code must provide values for the service attributes (sm\_service) listed in the following table.

Attribute	Description	Possible Values
containment_relation	The set of supported relations from which associations can be created.	SlmMonitors SlmWatchesContainer MaintenanceScheduledBy
name	The model name of the service.	Text string, up to 256 characters
Criticality	The impact severity of the alarm relative to other current alarms. A higher Criticality value contributes to a higher impact severity value for alarms in OneClick.	10 = Low 15 = Medium Low 20 = Medium 25 = Medium High 30 = High
AttrToWatch	The attribute monitored on the service resource models.  The AttrToWatch value should be consistent with the MonitorPolicy_ID. For example, if you specify AttrToWatch="Condition", you should specify a MonitorPolicy_ID for a Condition Policy (1-5).	Condition RM_Condition(service health) Contact_Status Port_Status LatestErrorStatus (Response Time)
MonitorPolicy_ID	The ID of a specific monitor policy as defined on the GlobalConfig model type. The ID should be consistent with the AttrToWatch value.	1 - 21 (CA Spectrum default) 1000 - n (User-defined)
Generate_Service_Alarms	Determines whether the SM_Service model generates alarms upon a change in service health.	True or False
Special_Cause_List	The alarm cause codes to include in an alarm type exemption list for a service.	Text string in the form of comma separated alarm causes or ranges separated with a hyphen (–)
Cause_List_Control	The integer that defines which alarm types affect or do not affect a service.	0=Unused (Ignore Cause) 1=Inclusive (Caused By) 2=Exclusive (Not Caused By)

### More information:

[Policy ID Mappings](#) (see page 183)

## Monitor Resource Monitor Attributes (SM\_AttrMonitor)

When creating service management models with Modeling Gateway, the XML code must provide values for the monitor resource monitor attributes (SM\_AttrMonitor) listed in the following table.

Attribute	Description	Possible Values
containment_relation	The set of supported relations from which associations can be created.	SlmMonitors SlmWatchesContainer
name	The model name of the resource monitor.	Text string, up to 256 characters
AttrToWatch	The attribute monitored by the resource monitor. The AttrToWatch value should be consistent with the MonitorPolicy_ID. For example, if you specify AttrToWatch="Condition", you should specify a MonitorPolicy_ID for a Condition Policy (1-5). <b>Note:</b> See <a href="#">Policy ID Mappings</a> (see page 183) for more information.	Condition RM_Condition (service health) Contact_Status Port_Status LatestErrorStatus (Response Time)
MonitorPolicy_ID	The ID of a specific monitor policy as defined on the GlobalConfig model type. It should be consistent with AttrToWatch value. <b>Note:</b> See <a href="#">Policy ID Mappings</a> (see page 183) for more information.	1 - 21 (CA Spectrum default) 1000 - <i>n</i> (User-defined)
Special_Cause_List	The alarm cause codes to include in an alarm type exemption list for a resource monitor.	Text string in the form of comma separated alarm causes or ranges separated with a hyphen (-)
Cause_List_Control	The integer that defines which alarm types affect or do not affect a resource monitor.	0=Unused (Ignore Cause) 1=Inclusive (Caused By) 2=Exclusive (Not Caused By)

## Customer Group Attributes (SM\_CustomerGroup)

When creating service management models with Modeling Gateway, the XML code must provide values for the customer group attributes (SM\_CustomerGroup) listed in the following table.

Attribute	Description	Possible Values
containment_relation	The set of supported relations from which associations can be created.	Groups_Customers
name	The model name of the customer group.	Text string, up to 256 characters

### More information:

[Example: Define a Customer and a Customer Group](#) (see page 137)

## Customer Attributes (SM\_Customer)

When creating service management models with Modeling Gateway, the XML code must provide values for the customer attributes (SM\_Customer) listed in the following table.

Attribute	Description	Possible Values
containment_relation	The set of supported relations from which associations can be created.	SlmUses
name	The model name of the customer.	Text string, up to 256 characters
CustomerField4	Address Line 1	Text string, up to 256 characters
CustomerField5	Address Line 2	Text string, up to 256 characters
CustomerField6	City, State, Postal Code	Text string, up to 256 characters
CustomerField7	Country	Text string, up to 256 characters
CustomerID	Any identification number.	Alpha-numeric string

Attribute	Description	Possible Values
Criticality	The impact severity of the alarm relative to other current alarms. A higher Criticality value contributes to a higher impact severity value for alarms in OneClick.	10 = Low 15 = Medium Low 20 = Medium 25 = Medium High 30 = High
Contact_Name	The person associated with the customer model.	Text string, up to 256 characters
Contact_Title	The contact person title.	Text string, up to 256 characters
Email_Address	The contact person email address.	Text string, up to 256 characters
Phone_Number	The contact person phone number.	Text string, up to 256 characters
Mobile_Phone_Number	The contact person mobile phone number.	Text string, up to 256 characters
Secondary_Contact_Name	The alternate contact person name.	Text string, up to 256 characters
Secondary_Contact_Title	The alternate contact person title.	Text string, up to 256 characters
Secondary_Phone_Number	The alternate contact person phone number.	Text string, up to 256 characters
Secondary_Mobile_Phone_Number	The alternate contact person mobile phone number.	Text string, up to 256 characters
Secondary_Email_Address	The alternate contact person email address.	Text string, up to 256 characters

**More information:**

[Example: Define a Customer and a Customer Group](#) (see page 137)

## SLA Attributes (SM\_SLA)

When creating service management models with Modeling Gateway, the XML code must provide values for the SLA attributes (SM\_SLA) listed in the following table.

Attribute	Description	Possible Values
containment_relation	The set of supported relations from which associations can be created.	SlaPeriod SImHasGuarantee SImGuarantees



Attribute	Description	Possible Values
name	The model name of the SLA.	Text string, up to 256 characters
SLA_Control	Specifies whether the SLA is active during the current SLA period or becomes active at the onset of the next period.	0 (inactive until next period) or 1 (active)
SLA_ExpirationDate	The UNIX timestamp, measured as the number of seconds from January 1, 1970.	For example, the value 1514696400 – Dec 31, 2017
SLA_Notes	Any text notes about the SLA.	Text string, up to 256 characters
SLA_Description	Any text description of the SLA.	Text string, up to 256 characters

**More information:**

[Examples: Create a Guarantee for an SLA](#) (see page 135)

[Example: Define an SLA](#) (see page 136)

## Guarantee Attributes (SM\_Guarantee)

When creating service management models with Modeling Gateway, the XML code must provide values for the guarantee attributes (SM\_Guarantee) listed in the following table.

Attribute	Description	Possible Values
containment_relation	The set of supported relations from which associations can be created.	SMIsMeasuredBy
name	The model name of the guarantee.	Text string, up to 256 characters
GuaranteeControl	Specifies whether the guarantee is active or inactive during the current period.	0 (inactive) or 1 (active)
GuaranteeType	Specifies whether the guarantee monitors service availability or performance (response time).	0 (availability) or 1 (performance)

Attribute	Description	Possible Values
ServiceHealthType	The type of service health time that is accumulated by the guarantee. An availability guarantee can accumulate both down and degraded time. A performance guarantee accumulates only degraded time.	1 (Down) or 2 (Degraded)
WarningThreshold	The number of seconds of outage time allowed per period before a warning alarm is issued.	0 - <i>n</i>
WarningThresholdPercent	The percentage of outage time allowed before a warning alarm is issued.	0 - 100%
ViolationThreshold	The number of seconds of outage time allowed per period before a violation occurs.	0 - <i>n</i>
ViolationThresholdPercent	The percentage of uptime per period below which a violation occurs.	0 - 100%
GuaranteeNotes	Any text notes about the guarantee.	Text string, up to 256 characters
GuaranteeDescription	A text description of the guarantee.	Text string, up to 256 characters
MOT_Threshold	Maximum outage time in seconds	0 - <i>n</i>
MTBF_Threshold	Mean time between faults in seconds	0 - <i>n</i>
MTTR_Threshold	Mean time to repair in seconds	0 - <i>n</i>

**More information:**

[Examples: Create a Guarantee for an SLA](#) (see page 135)

## Schedule Attributes (Schedule)

When creating service management models with Modeling Gateway, the XML code must provide values for the schedule attributes (Schedule) listed in the following table.

Attribute	Description	Possible Values
name	The model name of the schedule. <b>Note:</b> CA Spectrum renames the schedule name that you provide.	Text string, up to 256 characters
SCHED_Recurrence	Specifies when the schedule is implemented.	1 = Always (24 x 7) 2 = Daily 3 = Weekly 4 = Monthly 5 = Yearly
SCHED_Start_Hour	The hour the schedule starts.	0 - 23
SCHED_Start_Minute	The minute the schedule starts.	0 - 59
SCHED_Start_DoW	The day of the week the schedule starts	0 - 6
SCHED_Start_DoM	The day of the month the schedule starts.	1 - 31
SCHED_Start_Month	The month of the year the schedule starts.	0 (Jan.) - 11 (Dec.)
SCHED_Start_Year	The year the schedule starts. Entering 0 starts the schedule in the current year.	0
SCHED_Start_MoY	The month of the year the schedule starts. Entering 0 starts the schedule in the current month.	0
SCHED_Description	The description of the schedule.	Text string, up to 256 characters
SCHED_Duration	The duration the schedule is in effect in seconds.	0 - <i>n</i>
SCHED_Recurrence_Multiplier	The number of times the schedule is implemented.	1 - <i>n</i>
SCHED_Daily_Repeat_Limit	The number of consecutive days to repeat a daily schedule at the start of each recurrence period. Applicable to Weekly, Monthly, or Yearly recurrence only.	0 - <i>n</i>



# Chapter 7: Monitoring Service Management Components with the Service Dashboard

---

The Service Dashboard is Service Manager dedicated operational and administrative console. The Service Dashboard includes many of the same operational features as the OneClick Console, but focuses purely on service management components. The service dashboard offers an at-a-glance view to the real-time status of services, SLAs, and customers. Using Service Dashboard, you can also view history outage information, outage trends and summarized service availability.

This section contains the following topics:

- [The Service Dashboard](#) (see page 149)
- [Open the Service Dashboard](#) (see page 151)
- [Topology and List Views in the Contents Panel](#) (see page 153)
- [Explorer Folders and Topology Icons](#) (see page 153)
- [Status Indicators](#) (see page 154)
- [Access Information about a Service Management Component](#) (see page 155)
- [Service Dashboard Interface Management](#) (see page 157)
- [Locate Service Management Components](#) (see page 158)
- [Print Dashboard Views](#) (see page 159)
- [Export Dashboard Views](#) (see page 160)
- [Use the Service Dashboard Editing Tools](#) (see page 160)
- [Service Outage Management](#) (see page 162)

## The Service Dashboard

The Service Dashboard provides a service-centric view of your service management environment. Unlike the OneClick Console, which enforces user security at the data level, the Service Dashboard shows only the service, SLA, and customer models to which you have security access. If you do not have access to a specific model, you can still see explorer and topology icons in the OneClick Console. Within the service dashboard, if you do not have access to a model, that model is absent from all dashboard views. In some circumstances, you can allow the CA Spectrum Administrator to provide the service dashboard to specific CA Spectrum users to view only the appropriate service and SLA components.

Using the Service Dashboard, you can navigate to your service management environment through the dashboard explorer and topology panels. Component detail information for services, SLA, and customer models is equivalent to that available in OneClick with the addition of some new panels which display outage history and SLA/Guarantee trend information.

Unique to the Service Dashboard is the service topology view. The semi-customizable topology icons indicate service health. The topology icons are expandable and provide navigation to lower layer services. You can edit and annotate the topology view for those service hierarchies you create.

**More information:**

[OneClick Licenses and Service Manager Privileges](#) (see page 16)

## Open the Service Dashboard

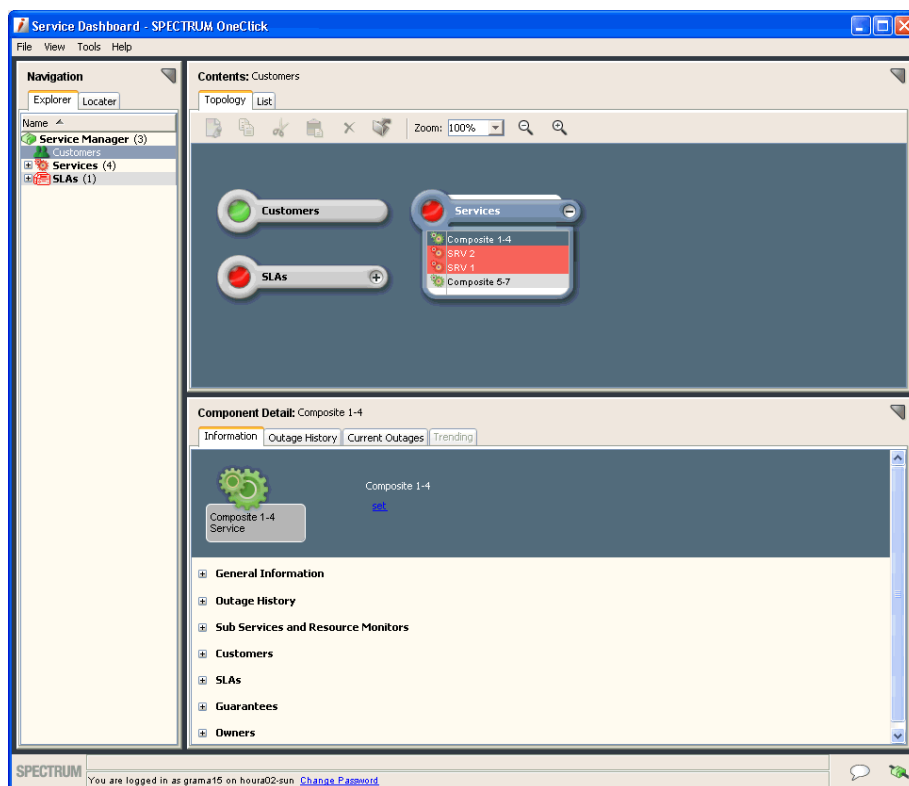
You can use various methods to open the Service Dashboard.

**Note:** You must have Service Manager license privileges to access the Service Dashboard.

**Follow these steps:**

1. Take *one* of the following steps:
  - Click the Service Dashboard link on the OneClick home page.
  - Click Tools, Utilities, Service Dashboard, from the main menu.
  - Right-click Service Manager in the Navigation panel and select Utilities, Service Dashboard in the OneClick Console.

The following image shows the areas of the Service Dashboard you work with to monitor service components.



The Service Dashboard interface includes three main information panels that you work with to monitor your organizations services management components:

### **Navigation Panel**

Displays components in a hierarchical folder structure. It provides the following options:

#### **Explorer tab**

Lets you select the component that you want to view in the Contents panel and the Component Details panel. It groups components by services, SLAs, and customers. The explorer tab in the service dashboard condenses all landscapes into a single tree. You can notice that services which reside on different landscapes are all organized into the same services folder. This tab allows you to view your service management implementation without consideration of how many SpectroSERVER are deployed.

#### **Locater tab**

Lets you search for the services, SLAs, or customers you want to view in the Contents panel. For example, can specify a particular component name or you can specify all components from a service management component category.

### **Contents Panel or Topology Panel**

Displays summary information about the status of components you specify in the Navigation panel. You can select Topology and List views of service management components that are selected in the Navigation panel. The panel topology view provides basic editing tools that you can use to arrange component icons, create basic shapes in the view, and annotate the view.

**Note:** Only the topology of user created hierarchies can be edited. Specifically the topology of the top tier folder that is named Services cannot be edited. This folder is comprised of multiple landscapes, and is shared among users. Given the variability in user access it would not be practical to allow topology editing to the top tier services topology.

### **Component Detail Panel**

Displays detailed information about components that are selected in the Navigation or Contents panels. Tables and sub-view available in the Component Detail panel contains service configuration information and real-time and historical outage information.

### **More information:**

[OneClick Licenses and Service Manager Privileges](#) (see page 16)

[View Outage History](#) (see page 163)



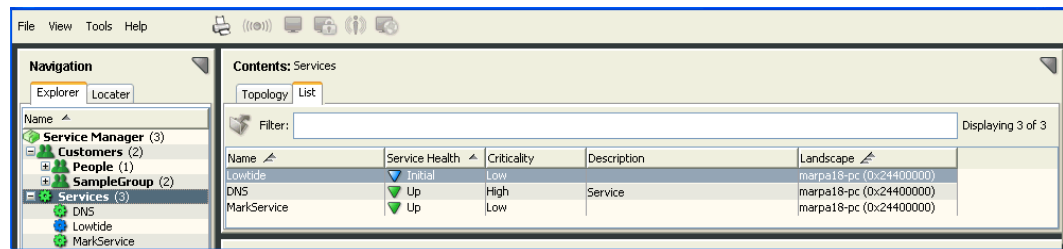
## Topology and List Views in the Contents Panel

The Contents Panel contains a Topology and List view. The topology view provides expandable icons which indicate the real-time status of each service management component. The List view provides a table of service component models and specific attribute data. When a model is selected in the navigation panel, both the List and Topology views show the selected models children. If the selected model has no logical children, the List and Topology views show the selected model as contained by its parent.

**Note:** The List view is the default view for component results that are found through the Locator tab. You can specify a List view for components that are selected from the Explorer tab.

The Component Details panel displays comprehensive information about the component that is selected in the Contents panel.

To display a List view of a service management component group that is selected in the Explorer tab, click the List tab in the Contents panel, as shown in the following image:



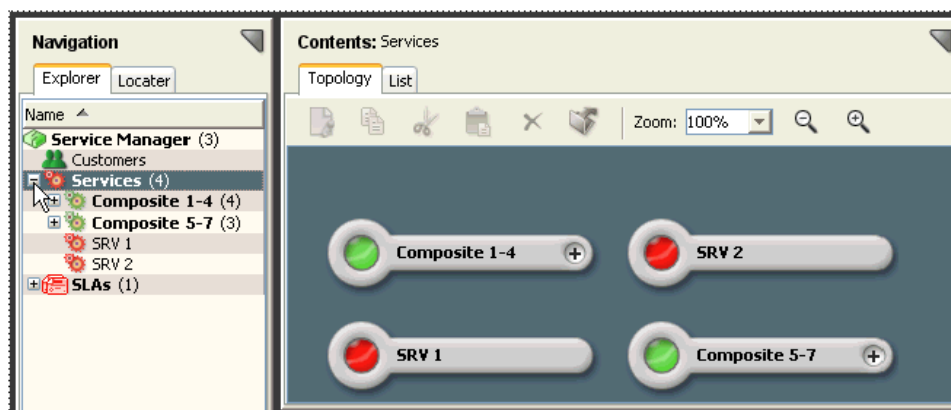
## Explorer Folders and Topology Icons

The service dashboard does not display the service hierarchy about the landscapes where the models reside. Within the service hierarchy, you can see only the parent/child relationships of their service models. Within the OneClick Console, a service can appear as a top tier service directly under the service manager model on its landscape. If the service is a child of a service on another landscape, it can be seen in the dashboard only as a child, and not as both a child and top tier service. This behavior has changed from earlier releases of Service Manager in which the service would have appeared both directly under the Services folder and a child of the other service.

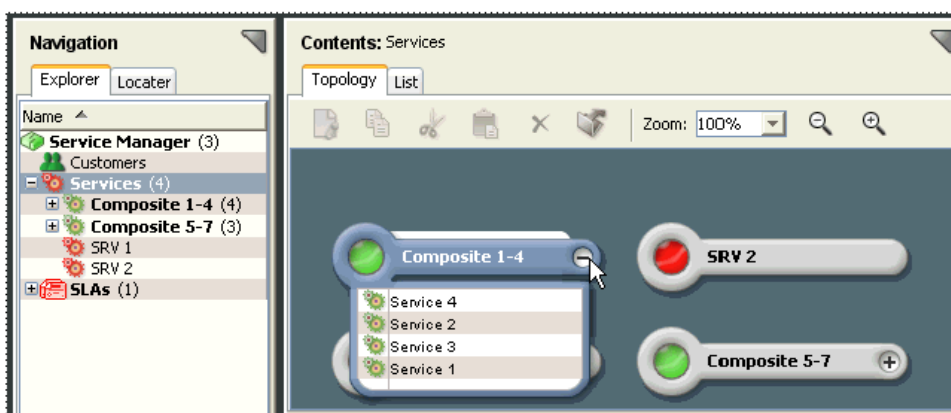
If a service management component displayed in the Explorer tab of the Navigation panel has a (+) next to it, this behavior indicates that the model or folder has one of more children. If you select a model, the List or Topology view displays the child models.

Icons within the topology view can also have (+) displayed in the icon itself. This behavior indicates that the model has children, click the plus sign (+) to expand and show a table of the associated child models.

For example, click the plus sign (+) next to a folder in the Navigation panel or on an icon in the Contents panel, or double-click a folder or icon:



For example, click the plus sign (+) on an icon in the Contents panel and then click a component from the drop-down list:



## Status Indicators

The top-level service management component icons—Services, SLAs, and Customers—indicate the most severely affected corresponding top-level category for each landscape. If the Services icon is red, it indicates that the service manager model (Services) in at least one landscape has a Condition of critical.

The following table describes service component icon colors, and the associated state attribute values:

Color	Service	SLA	Customer
Green	Normal	Unaffected	Not Impacted

Color	Service	SLA	Customer
Yellow	Slightly Degraded	Compliant	Slightly Impacted
Orange	Degraded	Warned (at risk)	Significantly Impacted
Red	Down	Violated	Severely Impacted

## Access Information about a Service Management Component

When you select a service management component from the Navigation panel or the Contents panel, the Service Dashboard Component Detail panel displays detailed information about it.

### Follow these steps:

1. [Open the Service Dashboard](#) (see page 151).
2. To access information in the Component Detail Panel, navigate to a model from either the Navigation or Contents views.
3. When the model is selected the Component Detail panel contains the following tabs:

#### Information tab

Displays detailed information about any service management component (customers, services, SLAs, guarantees). Contains a number of sub-view each with detailed configuration information or historical data.

#### Outage History tab

Displays the [outage history](#) (see page 163) for the last 31 days for the selected service.

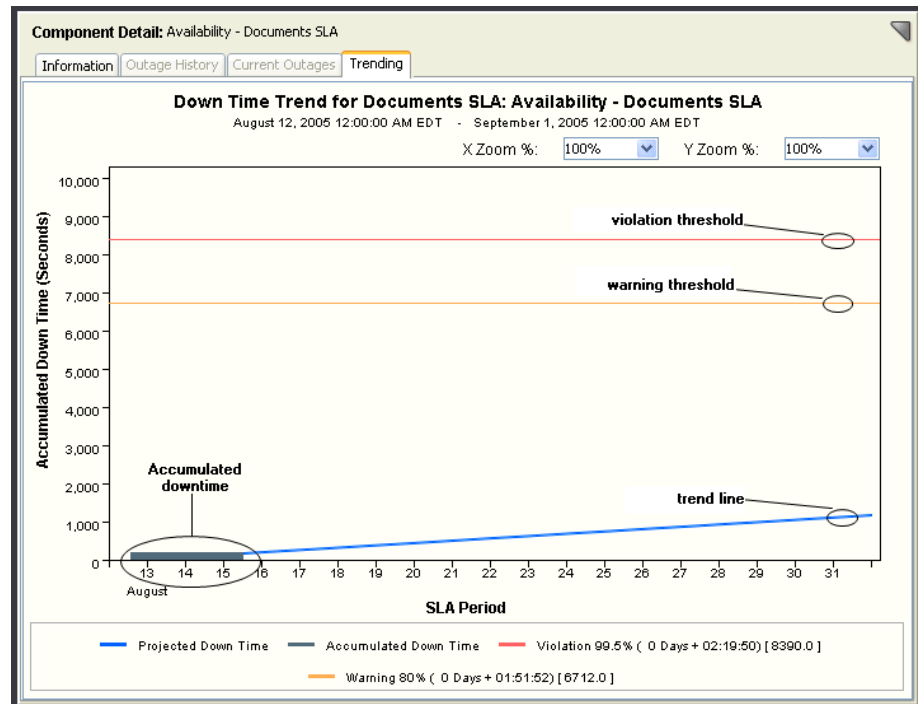
#### Current Outages tab (service and customer models only)

Displays information for any [current service outage](#) (see page 162), including the cause of the current outage, assignment, trouble ticket ID, and status note of root cause alarm.

### Trending tab (SLAs and guarantee models only)

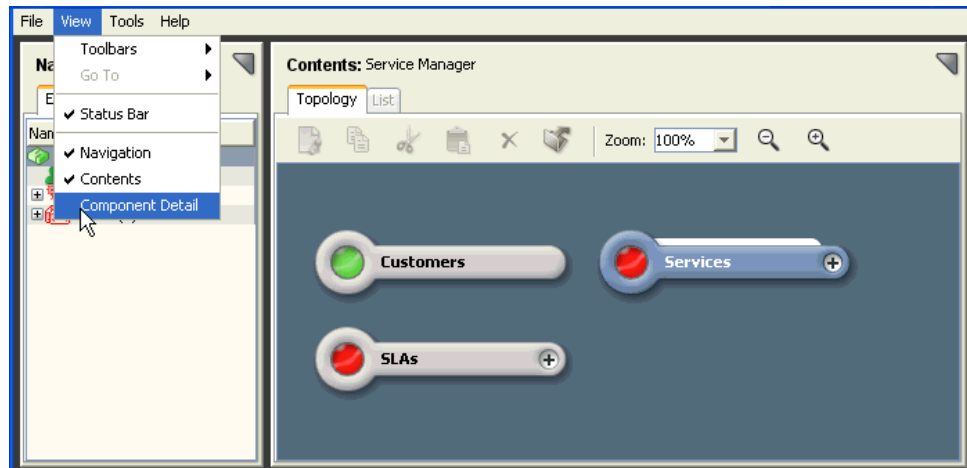
Displays accumulated outage time for a guarantee and a trend line to indicate the probability for the guarantee to become warned or violated within the current SLA period.

The following shows an example of a trend chart:



## Service Dashboard Interface Management

To show and hide Service Dashboard panels and the Status bar, select the interface component to display from the View menu, or clear the interface component to hide. The following image shows that the Component Detail panel has been deselected and is not included in the Dashboard interface:



To dock and undock Dashboard panels, which include the Navigation, Content, and Component Detail panels and Component Detail information panels, click the Dock/Undock icon. The following image shows an example of Docking/Undocking icon locations (circled). An undocked panel includes many of the same interface controls and options available from the main Service Dashboard.



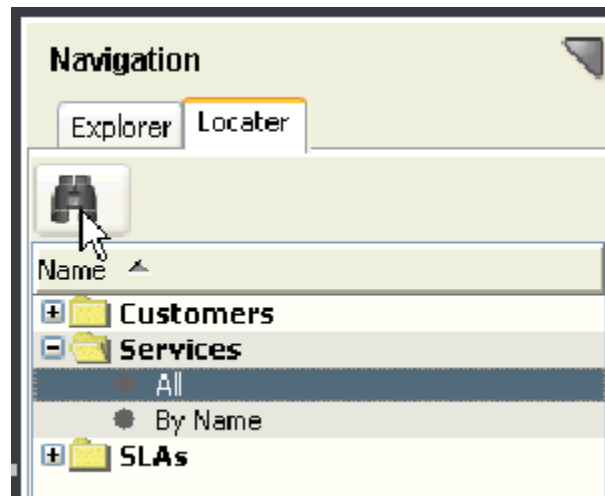
## Locate Service Management Components

You can use the Dashboard Locator tab to display those service management components you specify, in the Contents panel and Component Detail panel.

**Follow these steps:**

1. [Open the Service Dashboard](#) (see page 151).
2. Click the Locator tab in the Navigation panel.
3. Expand the customers, services, or SLAs folder from which you want to locate a component.
4. Specify whether to locate all components or a particular component from the selected component category:

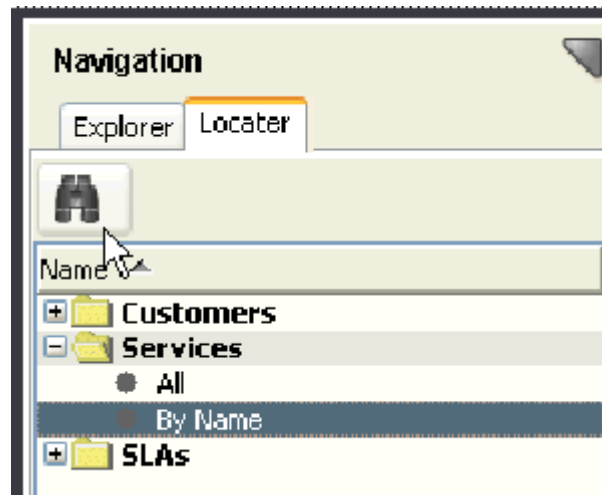
- Select All and click  to locate all components.



The Select landscape to search dialog appears.

Specify the landscape that you want to search and click OK.

- Select By Name and click  to locate a specific component.



The Search dialog appears.

Enter the component name in the Model Name field and click OK.

**Note:** To specify which landscapes to search, click the Landscapes button to open the Select Landscapes to Search dialog.

Results are displayed in the Contents panel.

## Print Dashboard Views

Service Dashboard lets you print content from the Explorer and Locator tabs. You can also print the content from the Topology and List tabs. The Results view in the Contents panel and the Information view of the Component Details panel can be printed.

### Follow these steps:

1. Click File, Print.

The Print dialog appears.

2. Select the content that you want to print, and click OK.


## Export Dashboard Views

Service Dashboard lets you save content from the Contents panel in multiple formats, using the export function.

You can export to a PNG (portable network graphics) format from the Explorer tab Topology view.

You can export to CSV (comma-separated values - spreadsheet compatible), text, and HTML formats from the Explorer tab List view and the Locator tab Results view.

### Follow these steps:

1. Click .  
The Save As dialog appears.
2. Select an available format and click Save.  
The dashboard view is exported.

## Use the Service Dashboard Editing Tools

The Service Dashboard editing tools let you customize the topology views for Service, SLAs, and Customer Groups. Only the topology views of user created models can be edited. You cannot edit the topology views of the top tier service management folders.

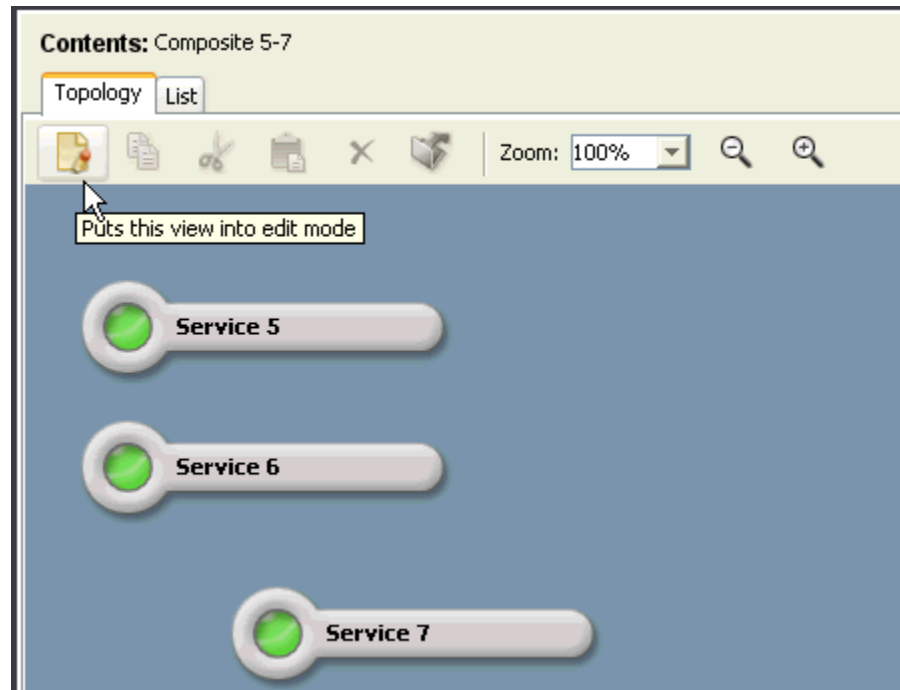
You can perform the following tasks:

- Arrange component icons.
- Create basic shapes (rectangles, ovals) and lines.
- Enter annotations.
- Modify annotation text color, font, style, and size.
- Modify the topology view background color, grid dimensions (which you can use to align and resize shapes and lines), and size.



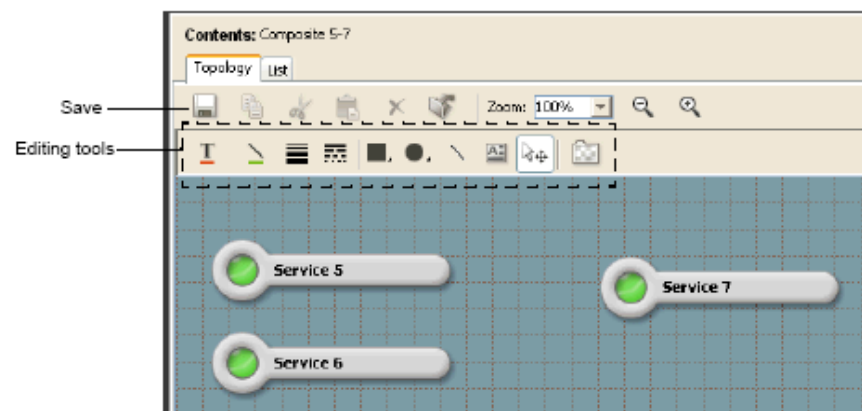
**Follow these steps:**

1. Click the Edit Mode icon in the Content panel Topology view, as shown in the following image:



The Topology view changes to editing mode and the editing toolbar appears. You can invoke a description of an editing tool by holding the cursor over the icon for the tool. The editing tools are a subset of tools available for editing the OneClick topology, and function in the same mapped.

Icon placement can be done either in tiled format or in custom format by selecting and dragging icons.



2. Edit the view as required and click Save.

## Service Outage Management

Service dashboard provides various options for viewing current or historical service outages. From the current outage tab you can view the following outage details:

- The health or severity of the outage
- The start time of the outage
- The resource or resources that are contributing the outage
- Any trouble shooter assignment or notation added to the outage

You can view and edit service outages from the Outage History tab panel or the Outage History sub-view of the Information tab panel.

The following information is available from the outage history table:

- Start time, end time, and duration of all outages.  
**NOTE:** With Service Manager r9.2, the outage limitation of 31 days has been removed. You can now specify the date range of all outages.
- The resource or resources that caused each outage
- The name of the person that is assigned to troubleshoot the current service outage
- The status of an outage (unplanned, planned, or exempt)

The Outage History tab panel also offers a pie chart depicting service health or historical availability for the past 31 days.

The service outage editor is available from either the Outage History tab panel of the outage history sub-view. The outage editor allows user to add outage note or change the status of an outage by marking it as exempt or planned.

## View Current Outages

Service Dashboard lets you view information about a service that is in an outage state.

**Follow these steps:**

1. [Open the Service Dashboard dialog](#) (see page 151).
2. Select the service with the outage that you want to review.
3. Click the Current Outage tab in the Component Detail panel.

The Component Detail panel displays information about the current outage, including the resource or resources that are contributing to the outage.

**More information:**

[Access Information about a Service Management Component](#) (see page 155)

## View Outage History

Service Dashboard lets you view information about all past and current outages for a service over the last 31 days.

**Follow these steps:**

1. [Open the Service Dashboard](#) (see page 151).
2. Select a service that you want to review.
3. Click the Outage History tab in the Component Detail panel.

The Component Detail panel displays information about the service outage history, including a chart with a summary of outage information and an itemized list of recent outages.

The panel also displays Outage Details pertaining to the resources which contributed to the outage.

**More information:**

[Access Information about a Service Management Component](#) (see page 155)

## Service Outages

A service outage can have one of three potential status values: Unplanned, Planned, or Exempt. Usually the status of a service outage influences only how the historical record for that outage can be interpreted.

Most service outages have the status of Unplanned unless they are edited by the user. Unplanned outage time counts against the historical availability of a service model. It can be recorded if an SLA has associated one of more guarantees to the service.

You can use the service outage editor to change the status of an Unplanned outage to either Planned or Exempt. A status of planned indicates that the outage was expected to happen, and can be considered a maintenance outage. A status of Exempt indicates that the outage was not expected, but the cause or nature of the outage is such that the outage time should not be counted against the historical availability of the service. Outages which have a status of planned or exempt does not contribute time to SLAs.

Editing an outage does not change the real-time status of a service. If a service is Down and the user marks the outage as exempt the service remains down, as that is the true health of the service. Even though the real-time health of the service is unchanged and if a guarantee records time for the outage, you can remove the time from the guarantee that can change the status of an SLA.

### Edit a Past or Ongoing Outage Status

You can edit the status of an ongoing or historical outage.

Consider the following information before editing a past or ongoing outage status:

- If an ongoing outage is edited, the outage status persists during the outage regardless of changes to the resources that contribute to the outage. Therefore, if the outage is marked as exempt, it remains exempt even if the resources contributing to the outage change. The current outage status can be seen in the Information tab for any service model.
- Outage time that is exempted or marked as planned is subtracted from the outage time tallied by a guarantee. This means that the status of an SLA that includes the guarantee changes accordingly. For example, an SLA with a status of violated resulting from a service outage that exceeded an availability guarantee threshold is restored to compliant when the outage is exempted.

#### Follow these steps:

1. [View Outage History](#) (see page 163) tab or open the Outage History sub-view of the Information tab.
2. Expand the Recent Outages subview in the Component Detail panel and select the outage that you want to edit from the Recent Outages list.

**Note:** Outages with health values of Down, Degraded or Slightly Degraded can be edited. You can also notice periods of Maintenance time, Initial time, and Loss of Management time in the outage table.

3. Click Outage Editor.

The Edit Service Outage dialog appears.

4. Select an outage type for the service outage from the Set Outage Type of the drop-down list. The following options are available:

#### Unplanned

Outages count against service availability.

#### Planned

Outages do not count against service availability.

### **Exempt**

Outages are the unplanned outages that you have determined and must not count against service availability.

5. (Optional) Enter comments in the Notes/Reason field for the outage you changed. You can also enter comments for any outage regardless of whether you changed its status.
6. Click Save.

The login name of the user who edited the outage is recorded and is displayed in the outage history table and in reports.

The outage status is edited.

**Note:** You can also edit outages with the Affected Services Editor in Report Manager. For more information, see the *Report Manager Installation and Administration Guide*.

### **More information:**

[View Outage History](#) (see page 163)



# Chapter 8: Monitoring Service Management Components with Unicenter Management Portal

---

This section contains the following topics:

[About the Service Level Manager Portlet](#) (see page 167)

[Publish the Service Level Manager Portlet in UMP](#) (see page 168)

[View Service Information](#) (see page 169)

[View SLA Information](#) (see page 170)

[View Customer Information](#) (see page 170)

[Open the OneClick Console and the Service Dashboard](#) (see page 171)

[Apply and Manage Layouts](#) (see page 171)

## About the Service Level Manager Portlet

The Service Level Manager portlet provides summary status information about services, SLAs, and customers to Unicenter users with security access to Service Manager models. The portlet can be incorporated into the Unicenter Management Portal (UMP).

The portlet displays information about services, SLAs, and customers in a customizable tabular format. The portlet also provides context-sensitive links to the OneClick Console and Service Dashboard where you can view more detailed information about the services, SLAs, and customers.

## Publish the Service Level Manager Portlet in UMP

Before viewing and accessing resources that are provided by the Service Level Manager portlet, first publish the portlet in UMP. For more information, see *UMP Documentation*.

**Follow these steps:**

1. Click the Knowledge tab and select Publish File.
2. Enter the URL for the Service Level Manager portlet in the Content box.  
For example: `http://abcde-sun/spectrum/slm`
3. Enter a title for the Service Level Manager portlet and click OK.  
The Library tab displays the portlet title.
4. Click the portlet title to open the portlet.  
You are prompted to log in with a CA Spectrum user name and password. After you log in, the portlet appears.
5. Click the Work Place tab and select the workplace where you want to add the portlet.
6. Select Add Content, select the Service Level Manager portlet name, add it to the desired column, and click OK.  
The portlet is published in UMP.



## View Service Information

You can use the Service Level Manager portlet to view service information.

### Follow these steps:

1. Click the Services tab.

The table displays status information about service models, as shown in the following image:

Services SLAs Customers		
Service Level Manager: Services		
OneClick		Service Dashboard
Name	Service Health	Criticality
<a href="#">All Pings 1</a>	Down	Low
<a href="#">All Pings 2</a>	Down	Low
<a href="#">All Pings 3</a>	Down	Low
<a href="#">Joes Service</a>	Degraded	Low
<a href="#">Service 1</a>	Degraded	Low
<a href="#">Sir LP3 Service</a>	Degraded	Low
<a href="#">spm RT &gt;80</a>	Degraded	Low
<a href="#">ABC Service</a>	Up	Low
<a href="#">Service 1B</a>	Up	Low

2. To view context-sensitive views of a service in OneClick and Service Dashboard, perform one of the following actions:
  - Click a service Name to view a detailed information view of the service in OneClick Console.
  - Click the Service Health indicator for a service to view information about the service in Service Dashboard.

## View SLA Information

You can use the Service Level Manager portlet to view SLA information.

### Follow these steps:

1. Click the SLAs tab.

The table displays status information about SLA models, as shown in the following image:

Service Level Manager: SLAs			
Name	Status	Start Time	End Time
AndreasT SLA	Violated	Aug 21, 2006 12:00:00 AM GMT-04:00	Aug 22, 2006 12:00:00 AM GMT-04:00
rspm > 80 sla	Violated	Aug 16, 2006 12:20:27 PM GMT-04:00	Sep 1, 2006 12:00:00 AM GMT-04:00
LP3 Gold	Initial	Aug 16, 2006 10:42:47 AM GMT-04:00	Sep 1, 2006 12:00:00 AM GMT-04:00

Filter: All [Go] [Reset] Displayed 3 of 3

Displaying 1 - 3 of 3 items.

2. To view context-sensitive views of an SLA in OneClick Console and Service Dashboard, perform one of the following actions:
  - Click an SLA Name to view a detailed information view of the SLA in OneClick Console.
  - Click the Status indicator for an SLA to view information about the SLA in Service Dashboard.

## View Customer Information

You can use the Service Level Manager portlet to view customer information.

### Follow these steps:

1. Click the Customers tab.

The table displays status information about customer models, as shown in the following image:

Service Level Manager: Customers				
Customer Impact	Name	ID	Criticality	Primary Contact Name
None	Universal Widgets		Medium	George Rogers
None	Advanced Stuff		Low	Fred Snuffles
None	Customer A		Low	A
None	Customer B		Low	B
None	National Noise Producers		Low	Barney Rubble

Filter: All [Go] [Reset]

Displaying 1 - 5 of 5 items.

2. To view context-sensitive views of a customer in OneClick and Service Dashboard, perform one of the following actions:
  - Click a customer Name to view a detailed information view of the customer in OneClick Console.
  - Click the Customer Impact indicator for a customer SLA to view information about the customer in Service Dashboard.

## Open the OneClick Console and the Service Dashboard

You can open the OneClick Console and the Service Dashboard from any component view in the Service Level Manager portlet. Perform one of the following actions:

- Click the OneClick button.
- Click the Service Dashboard button.

## Apply and Manage Layouts

The Service Level Manager portlet lets you customize the type of information to include in the services, SLAs, and customers views. It lets you customize how you want to organize the information. You can also save multiple customized layouts for each view, edit all user-created layouts, export and import layout files, and remove layouts from the Service Level Management portlet.

The following image shows an example default layout for a service:

**Table Layout Configuration**

Layout Name:

**Available columns**

- Generate Service Alarms

**Column Order**

Show these columns in this order

- Name
- Service Health
- Criticality
- Description

Add -> <- Remove

**Sort Order**

Sort by: Service Health (Ascending)

then by: Name (Ascending)

then by: None (Ascending)

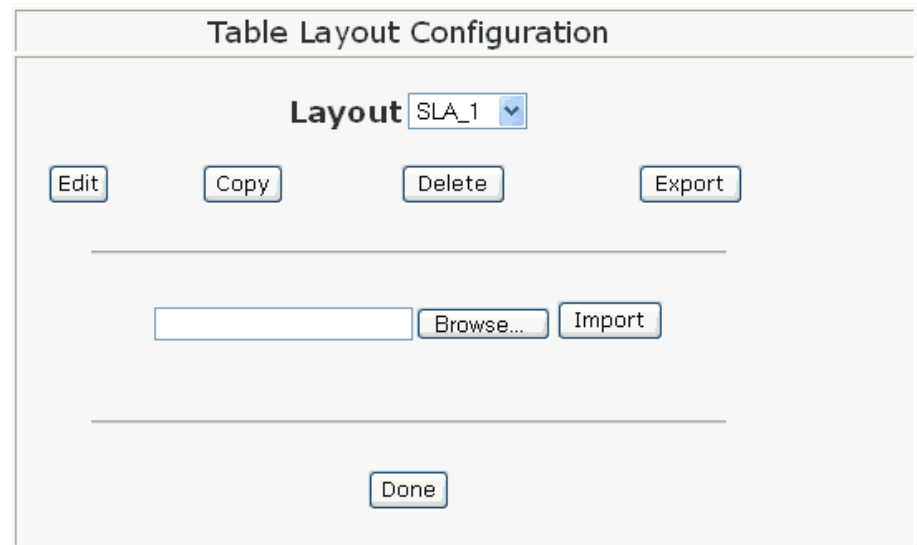
Items per Page: 10

Save Cancel

Perform the following actions to apply and manage layouts:

- To apply a layout to a component view, select it from the Configuration drop-down list. The table displays the columns and sort order that is specified for the layout.
- To create a layout for a component view, apply the default layout to the view, click Configure to open the Table Layout Configuration window, specify settings, and click Save.
- To manage a layout, apply the layout that you want to edit, copy, or delete to the current component view and click Configure.

The Table Layout Configuration window appears with available layout management options, as shown in the following image:



The screenshot shows a window titled "Table Layout Configuration". At the top, there is a label "Layout" followed by a dropdown menu currently displaying "SLA\_1". Below this, there are four buttons arranged horizontally: "Edit", "Copy", "Delete", and "Export". A horizontal line separates these buttons from the next section. In this section, there is a text input field, a "Browse..." button, and an "Import" button. Another horizontal line is below this section, and at the bottom center of the window is a "Done" button.

Select the following actions that you want to perform:

- Click Edit if you want to modify the layout.
- Click Copy if you want to create another version of the layout.
- Click Delete if you want to remove the layout from Service Level Management. You can export a layout to your hard drive before you delete it in case you want to re-use the layout in the future by importing it back into Service Level Management.
- Click Export if you want to save a copy of the layout to your hard drive (as a .prx file).
- Click Browse to locate a layout file on your hard drive and then click Import to add a layout to the list of available layouts.

# Chapter 9: Generating Service Manager Reports

---

You can generate various informative reports about services, SLAs, and customers with CA Spectrum Report Manager. This chapter briefly describes the types of service management reports you can generate and how to generate reports. An application administrator is responsible for managing access to reports. For more information, see the *Report Manager Installation, Administration Guide*, and *Report Manager User Guide*.

This section contains the following topics:

[Service and SLA Reports](#) (see page 173)  
[Outage Reports](#) (see page 175)  
[Inventory Reports](#) (see page 176)  
[Detailed Availability Reports](#) (see page 177)  
[Summarized Availability Reports](#) (see page 177)  
[Customer Reports](#) (see page 178)  
[SLA Status Reports](#) (see page 179)  
[Health Reports](#) (see page 181)  
[Generate Reports](#) (see page 182)

## Service and SLA Reports

### Report Folders

A number of sub folders are available within the Service and SLA package. These sub folders are used to organize reports by the type of data the report provides. Many reports contain status and outage information that can be produced for a specified time ranges. Other reports contain configuration information that is based on the current time. The follow sub folders are available for Service and SLA reports.

- Detailed Availability
- Summarized Availability
- Health
- SLA Status
- Inventory
- Customer

## Report Categories

Service Manager reports can be loosely categorized into two groups; Customer Facing and Administrative.

Although, a significant number of content overlap customer facing reports tend to be simpler than administrative reports. In some cases, a report can be designed specifically to be customer facing, in other cases a report may have parameters allowing it to be tailored for a specific customer.

Administrative reports tend to be detailed and are not specific to any customers services or SLAs. Administrative reports can show health or status values which would not be appropriate to expose to customers. Administrative reports can also show system-wide statistics which span the services of different customers.

For example, a service administrator can generate two sets of service availability and SLA reports for a given period. The Service Availability By Customer, and SLA Detail By Customer shows basic availability and SLA compliance for all of the customer services and SLAs. These customer facing reports can be distributed to the service customers. The administrator can also generate a Service Health and SLA Detail By Name report for their own use. These reports show more detail including lesser severity service outages, and SLA status. The administrator looks for services which experienced considerable degraded time, or SLAs which had a warned status. Using these more detailed reports can help the service administrator identify services or SLA which are at risk or otherwise unstable.

## Reporting Data

Reports that are found in the Service and SLA folder are based on data from the service manager-specific tables in the reporting database. Because service outage information is also displayed in the OneClick client. The service manager tables in the reporting database are updated in real time. You can run a service availability, service health, or SLA status report and can see the actual real-time status of that model.

## Report Structure and Exporting Reports

Most service and SLA reports have a similar structure. The main report includes summarized data, and may include various charts or graphs. More detailed data is available by navigating to sub reports. Report Manager supports exporting report data to other common formats, but only the main report data is exported.

Sub-report data is not available through export, if a specific user needs access to detailed sub-report data, privileges must be given to generate reports using Report Manager.

## Report Parameters

Many of the service manager reports use the standard report manager parameter for selecting time and data ranges to report on. For SLA reports the time that is specified may not completely encompass a single SLA period. However, the SLA report shows the complete period of data.

In addition, Service and SLA reports offer additional parameters that allow for the customization of report titles and chart titles. Some reports often parameter options allows you to specify the data to be displayed and how availability is calculated.

By default Service and SLA reports display data only for models which still exist in the CA Spectrum environment. Many reports provide a parameter that allows you to report on models that have been destroyed.

## Outage Reports

The following outage reports are useful for the service administrator. These reports provide insight into the worst performing services, the worst overall service outages, and the service resources that contribute the greatest amount of service outage time. All outage reports allow you to specify how the time range to report for, and the number of models or outages to summarize in the report.

### **Top N Worst Performing Services**

Displays the top N worst services in terms of total down time. User can select how many services to report on, the default is 10. The report contains a bar chart for up to ten services. The user can specify the units of the time axis for the bar chart to be in seconds, minutes, hours, or days. For each service listed in the main report a service availability sub report can be accessed by clicking the service name. This version of the service availability sub report shows only down time, and normal time.

### **Top N Worst Performing Services Including All Outage Types**

Displays the top N worst services in terms of total down time, but also displays summarized degraded and slightly degraded time. User can select how many services to report on, the default is 10. The report contains a bar chart for up to ten services. The user can specify the units of the time axis for the bare chart to be in seconds, minutes, hours, or days. For each service listed in the main report a service availability sub report can be accessed by clicking the service name. This version of the service availability sub report shows down, degraded and slightly degraded and normal time.

### **Top N Worst Service Outages**

Displays the top N worst service outages, and displays a summarized bar chart for the top 10 plus the rest. This report is useful for identifying specific outages that resulted in long periods of service outage time. If the environment does not experience long periods of service down time, this report cannot be revealing. You can specify the number of outages to display and the units of the outage chart. For each outage, you can navigate to a sub report to view the impacted service and any impacted customers.

### **Top N Worst Service Resources By Total Down Time**

Lists the service resources which contributed to the greatest amount of service down time. The calculation for service down time is cumulative, although, one resource is down for less actual time than another and its cumulative service impact can be greater. The default number of resources that are displayed is 50. From each resource, a sub report is available showing the specific outage times and impacted services. An outage impacting multiple services can be listed multiple times, and grouped by shading to show that it is really a single resource outage which led to multiple service outages. This report is useful for service administrators to understand if there are any particular service resources that contribute to high amounts of service outage time. Service administrators can replace these resources with more reliable products, or perhaps add redundancy such that the service impact of an individual resource is minimized.

## Inventory Reports

Inventory reports provide a snap shot of the current modeling in CA Spectrum. You can specify time ranges when running the report. It produces data based on the current time and configuration. Inventory reports show only service and SLA structure and not status. SLA Inventory reports display the configuration of specific SLA models and their guarantees. It is appropriate to provide a customer with an inventory report for their SLAs. This confirms that the customer understands the configuration of each SLA and the thresholds that are specified for each guarantee.

### **Service Inventory**

Lists all service models and their resources. It is a complex report and can be useful to export to a spread sheet. The tabular form of this report makes it difficult to decipher the service hierarchy as many services monitor other services. This report can be used to capture a snap shot of the overall service hierarchy which can be saved and referenced.

### **SLA Inventory by SLA Customer**

Displays a list of all of a customer SLA models and the configuration for each guarantee. It can be useful to distribute to customers so that they understand the configuration for each SLA. This report is also available in the Customer folder.

### **SLA Inventory by SLA Name**

Lists each user specified SLA model and its guarantee configurations. This report provides a quick way to review SLA configuration without having to navigate to each SLA through the OneClick interface.



## Detailed Availability Reports

Detailed availability reports provide information about the availability and outages of services in CA Spectrum organized by customers or service name. These reports are customer facing, each report displays a pie chart summarizing service availability, and lists outages which impacted the availability of the service. From each outage, a sub report is available showing the resource faults that contributed to the service outage.

**Note:** The resource detail is not available when the report is exported to another format.

### **Service Availability By Service Customer**

Produces a service availability report for all of a customer service models. This availability report display only down time and normal time. This report is also available in the Customer folder.

### **Service Availability By Service Name**

Produces a service availability report for each specified service. This availability report displays only down time and normal time.

### **Service Availability Variable Health Level**

Produces a service availability report for each specified service. Provides a parameter which specifies which service health types can be displayed and factored into the availability calculation. By default Down, Degraded and Slightly Degraded outage types are show, optionally the report could show Down only or Down and Degraded. It can be useful to produce this report for customers who are interested in seeing a service availability report which includes outages of lesser severity.

## Summarized Availability Reports

Summarized availability reports provide summary information about service availability and outages for a specified time range. These reports are customer facing, and can be useful to distribute to customers. Services are listed in tabular form, but provide a sub report that displays service availability. Sub report is not available if exporting to another format.

### **Service Summary By Service Customer**

Lists all of the customer service model with summarized availability statistics. Provides a link to a sub report which displays down and normal time. This report is also available in the Customer folder.

### **Service Summary By Service Name**

Lists all specified service models and summarized availability statistics. Provides a link to a sub report which displays down and normal time.

### **Service Summary Variable Service Health**

Lists all specified service models, and configurable availability statistics. Provides a parameter which specifies which service health types can be displayed and factored into the availability calculation. By default Down, Degraded and Slightly Degraded outage types are shown, optionally the report could show Down only or Down and Degraded. It can be useful to produce this report for customers who are interested in seeing a service availability report which includes outages of lesser severity.

## **Customer Reports**

Customer reports provide information about service customers, and the services and SLAs those customers are associated with.

### **Customer Detail**

Provides contact information and a list of managed services for a specific customer. The listed services include only the service models that are currently associated to the customer. This report does not include any status information.

### **Customer SLA Summary**

Provides a summary of customer SLAs and status for up to the six most recent SLA periods. This report is useful for the service administrator to review the status of a customer SLA for some recent periods. This report can be appropriate to distribute to customers, but it does display SLA warning status which is not available in the SLA Detail By Customer report.

### **Service Availability by Service Customer**

Produces a service availability report for all the customer service models. This availability report displays only down time and normal time. This report is also available in the Detailed Availability folder.

### **Service Summary by Service Customer**

Lists all of the customer service models with summarized availability statistics. Provides a link to a sub report which displays down and normal time. This report is also available in the Summarized Availability folder.

### **SLA Detail by Customer**

Displays the status for all of the customer SLA models for a specified amount of time. The time range may span multiple periods. For each guarantee listed, a sub report shows outages impacting the guarantee for the specific SLA period. This report can be distributed to customers allowing them to see the status of their SLA models. Remember that the sub report data for guarantee periods details are not available if this report is exported to another format. This report is also available in the SLA Status folder.

**SLA Inventory by SLA Customer**

Displays a list of all of the customer SLA models and the configuration for each guarantee. It is useful to distribute to customers so that they understand the configuration for each SLA. This report is also available in the Inventory folder.

**SLA Status Current and Recent by Customer**

Displays a table for all of the customer SLA models with the status for the current period, and any number of past periods that are specified by the user. For each SLA model listed, a set of guarantee detail sub reports is available by clicking the SLA name and paging through each sub report. The guarantee period detail report displays outages impacting the guarantee for the specific SLA period. This report can be appropriate to distribute to customers, however, it displays SLA warned status values which are not exposed in the SLA Detail by Customer report. This report is also available in the SLA Status folder.

**SLA Summary by Customer**

Displays table for all of the customer SLA models that are organized by SLA status and arranged by SLA period. For each SLA model listed, a set of guarantee detail sub reports is available by clicking the Period Details link and paging through each sub report. The guarantee period detail report displays outages impacting the guarantee for the specific SLA period. This report can be appropriate to distribute to customers, however, it displays SLA warned status values which are not exposed in the SLA Detail by Customer report. This report is also available in the SLA Status folder.

## SLA Status Reports

Various SLA status reports are available in the SLA Status folder. Many of these reports can be distributed to customers, but can remember that sub report data is not available in exported reports. Although some reports allow you to specify time ranges, an SLA report always be expressed in terms of one or complete SLA periods. For example, if the SLA period is monthly it does not make sense to generate a report for the previous day, as daily status has no bearing on a monthly SLA. If you generate a report for any portion of an SLA period, the report contains data for the complete period. If you generate a report for multiple SLA periods the report can be organized to display each period individually.

### **SLA Detail By Customer**

Displays the status for all of the customer SLA models for a specified amount of time. The time range can span multiple periods, and the report organizes each guarantee by period. This report displays only the compliant and violated status values. The SLA Status values of Unaffected, Compliant(Affected), and Warned are all displayed as Compliant in this report. For each guarantee listed, a sub report show outage impacting the guarantee for the specific SLA period. This report can be distributed to customers allowing them to see the status of their SLA models. Remember that the sub report data for guarantee periods details are not available if this report is exported to another format. This report is also available in the Customer folder.

### **SLA Detail By SLA Name**

Displays the status for all specified SLA models for a specified amount of time. The time range can span multiple periods, and the report organizes each guarantee by period. This report shows all SLA Status values including Compliant(Affected) and Warned which is not appropriate for customers. For each guarantee listed, a sub report show outage impacting the guarantee for the specific SLA period. Remember that the sub report data for guarantee periods details are not available if this report is exported to another format.

### **SLA Detail With Resource Outages**

Displays the status of an SLA for a specific period and includes detailed information about the service resources which led to any SLA impacting outages. This is a specialized report and is not appropriate for all users. If the SLA experienced many outages or impacted by many resources, the report can be large, and does not display correctly. Because this report transcends the service layer by displaying actual resource about SLA, it is not useful for most enterprise style services. This report can be useful in service provider environments where the service itself is essentially a single customer edge device or set of devices.

### **SLA Status Current and Recent**

Displays a table for all specified SLAs with the status for the current period, and any number of past periods that are specified by the user. For each SLA model listed, a set of guarantee detail sub reports is available by clicking the SLA name and paging through each sub report. The guarantee period detail report displays outages impacting the guarantee for the specific SLA period.

### **SLA Status Current and Recent By Customer**

Displays a table for all of the customer SLA models with the status for the current period, and any number of past periods that are specified by the user. For each SLA model listed, a set of guarantee detail sub reports is available by clicking the SLA name and paging through each sub report. The guarantee period detail report displays outages impacting the guarantee for the specific SLA period. This report can be appropriate to distribute to customers, however, it displays SLA warned status values which are not exposed in the SLA Detail by Customer report. This report is also available in the Customer folder.

**SLA Summary By Customer**

Displays a table for all of the customer SLA models organized by SLA status and arranged by SLA period. For each SLA model listed, a set of guarantee detail sub reports is available by clicking the Period Details link and paging through each sub report. The guarantee period detail report displays outages impacting the guarantee for the specific SLA period. This report can be appropriate to distribute to customers, however, it displays SLA warned status values which are not exposed in the SLA Detail by Customer report. This report is also available in the Customer folder.

**SLA Summary By Name**

Displays tables that are organized by status for all specified SLAs. Each status-based table is organized by period, and can contain entries for multiple SLA periods. For each SLA model listed, a set of guarantee detail sub reports is available by clicking the Period Details link and paging through each sub report. The guarantee period detail report displays outages impacting the guarantee for the specific SLA period.

**SLA Summary By Status**

Displays tables that are organized by status for all SLAs. Each status-based table is organized by period, and can contain entries for multiple SLA periods. For each SLA model listed, a set of guarantee detail sub reports is available by clicking the Period Details link and paging through each sub report. The guarantee period detail report displays outages impacting the guarantee for the specific SLA period.

**SLA Summary of Warned or Violated SLAs**

Displays table of all SLAs that are currently warned or violated. For each SLA model listed, a set of guarantee detail sub reports is available by clicking the Period Details link and paging through each sub report. The guarantee period detail report displays outages impacting the guarantee for the current SLA period. This report is useful for the service administrator to get a quick summary of all SLAs that are currently violated or at risk of becoming violated.

## Health Reports

A service health report provides information detailed service health information for service models and resource monitors for a specified period. The report differs from the availability report in that it includes all aspects of service health, including up, down, degraded, slightly degraded, and Maintenance time.

**Service or Resource Monitor Health by Name**

Displays a pie chart summarizing policy violation and incidental outage time. Lists all outages, and provides a sub report for policy violation outage which displays the resources that contribute to the outage. The outage detail is not available if the report is exported to another format. This report is the most comprehensive report available for a service administrator to understand the historical status for any service at any period.

## Generate Reports

Before you can access Report Manager and can generate Service Manager reports, you must have a Report Manager user account and the appropriate permissions (rights) to the Service and SLA report pack. Request both from your OneClick/Report Manager administrator.

**Note:** By default, Report Manager does not generate reports for services that have been deleted from CA Spectrum. To include deleted service information, select "True" for "Include Deleted SLAs" when you configure the report.

**Note:** Currently Report Manager does not enforce CA Spectrum model security for individual service or SLA models. Anyone with privileges for the Service and SLA content area can report on any service or SLA model.

**Follow these steps:**

1. Access Report Manager from the OneClick Console.
2. Click the On Demand Reports tab.
3. Select the Service and SLA folder.
4. Select the report category from which you want to generate a report. Report Manager displays a list of reports from the category you selected.
5. Select a report from the list.
6. Configure the report, and then click View Report.

The report is generated.

After you generate the report, you can select to save it as a favorite, which you can generate anytime without having to configure it again. You can also select to schedule the report to run on a one-time or on-going basis.

# Appendix A: Service Manager Policy Descriptions

---

This section contains the following topics:

[Policy ID Mappings](#) (see page 183)  
[Condition Value Sum Greater Than Or Equal](#) (see page 185)  
[Port Status Policies](#) (see page 186)  
[Condition Policies](#) (see page 187)  
[Response Time Policies](#) (see page 188)  
[Service Health Policies](#) (see page 189)  
[Contact Status Policies](#) (see page 190)

## Policy ID Mappings

The following table lists associations between watched attributes and monitor Policy IDs (1-21) for standard monitoring policies that are shipped with Service Manager.

**Note:** User-defined policies begin with ID value 1000 (and are incremented by 1, 1001, 1002, and so on).

To view user-defined Policy IDs, access the following link: <http://<server>:CAPortal/spectrum/slm/policyrep.jsp>.

Watched Attribute (AttrToWatch)	Monitor Policy ID (MonitorPolicy_ID)	Policy
Condition_Value	1	<a href="#">Condition Value Sum Greater Than Or Equal</a> (see page 185)
Condition	2	<a href="#">Condition Redundancy</a> (see page 187)
Condition	3	<a href="#">Condition High Sensitivity (Default Policy)</a> (see page 187)
Condition	4	<a href="#">Condition Low Sensitivity</a> (see page 187)
Condition	5	<a href="#">Condition Percentage</a> (see page 187)

Watched Attribute (AttrToWatch)	Monitor Policy ID (MonitorPolicy_ID)	Policy
RM_Condition	6	<a href="#">Service Health Redundancy</a> (see page 189)
RM_Condition	7	<a href="#">Service Health High Sensitivity</a> (see page 189)
RM_Condition	8	<a href="#">Service Health Low Sensitivity</a> (see page 189)
RM_Condition	9	<a href="#">Service Health Percentage</a> (see page 189)
Contact_Status	10	<a href="#">Contact Status Redundancy</a> (see page 190)
Contact_Status	11	<a href="#">Contact Status High Sensitivity</a> (see page 190)
Contact_Status	12	<a href="#">Contact Status Low Sensitivity</a> (see page 190)
Contact_Status	13	<a href="#">Contact Status Percentage</a> (see page 190)
Port_Status	14	<a href="#">Port Status Redundancy</a> (see page 186)
Port_Status	15	<a href="#">Port Status High Sensitivity</a> (see page 186)
Port_Status	16	<a href="#">Port Status Low Sensitivity</a> (see page 186)
Port_Status	17	<a href="#">Port Status Percentage</a> (see page 186)
LatestErrorStatus	18	<a href="#">Response Time Redundancy</a> (see page 188)
LatestErrorStatus	19	<a href="#">Response Time High Sensitivity</a> (see page 188)
LatestErrorStatus	20	<a href="#">Response Time Low Sensitivity</a> (see page 188)
LatestErrorStatus	21	<a href="#">Response Time Percentage</a> (see page 188)



**More information:**

[Policies and Watched Attributes](#) (see page 122)

[Monitor Resource Monitor Attributes \(SM\\_AttrMonitor\)](#) (see page 142)

[Service Attributes \(SM\\_Service\)](#) (see page 141)

## Condition Value Sum Greater Than Or Equal

Watched attribute: Condition\_Value

Default reason: The condition value has violated the allowable threshold.

This policy is different from the other standard Service Manager policies. Other policies include an attribute map, which defines the attributes that are monitored. This policy monitors the aggregate value of the Condition\_Value (0x1000b) attribute for all resources that are associated with a service.

**Rule Set**

- When the sum of the Condition\_Value attribute for all monitored resources is equal to or greater than the value of Red\_Threshold (0x10012), the service is down.
- When the sum of the Condition\_Value attribute for all monitored resources is equal to or greater than the value of Orange\_Threshold (0x10011), the service is degraded.
- When the sum of the Condition\_Value attribute for all monitored resources is equal to or greater than the value of Yellow\_Threshold (0x10010), the service is slightly degraded.

You can adjust the values for Red\_Threshold, Orange\_Threshold, and Yellow\_Threshold on the service model, or can adjust Value\_When\_Red (0x1000e), Value\_When\_Orange (0x1000d), or Value\_When\_Yellow (0x1000c) on the service component resource models to obtain the desired behavior.

**More information:**

[Policy ID Mappings](#) (see page 183)

## Port Status Policies

Watched attribute: Port\_Status

Default Reason: Bad Port Status

### Value Mapping

- Service Health = Down if the watched attribute value is down, disabled, or unreachable
- Service Health = Up if the watched attribute value is up

### Port Status High Sensitivity

- When any 1 resource is Down then the service is Down.

### Port Status Low Sensitivity

- When all resources are Down then the service is Down.
- When all resource(s) are Down then the service is Degraded.

### Port Status Redundancy

- When all resources are Down then the service is Down.
- When any 1 resource(s) is Down then the service is Slightly Degraded.

### Port Status Percentage

- When 75 percent of the resources are Down then the service is Down.
- When 50 percent of the resources are Down then the service is Degraded.
- When 25 percent of the resources are Down then the service is Slightly Degraded.

**Port Status - Automatically Configured**

PollPortStatus has been automatically configured for interface models which are added or removed from services or resource monitors. Verify the following scenarios for Port Status:

- If an interface model has a PollPortStatus of False (No), when the model is added to a service or resource monitor PollPortStatus can be set to True (Yes). (OneClick View Attributes which imply false are displayed as 'No' and true are displayed as 'Yes'.)
- If an interface model has a PollPortStatus of True, when the model is added to a service or resource monitor, no changes are made. When the model is removed from the service or resource monitor the PollPortStatus is left as true.
- There is a caveat to the functionality described earlier. In scenarios where the PollPortStatus is automatically updated from false to true, that information is only preserved for the lifecycle of the running SpectroSERVER. The PollPortStatus would not be restored to False when the interface is removed from the service, if the SpectroSERVER shutdown between the time the interface was added and removed.

**More information:**

[Policy ID Mappings](#) (see page 183)

## Condition Policies

Watched attribute: Condition\_Value

Default reason: Bad Condition

**Value Mapping**

- Service Health = Down if the watched attribute value is Critical or Suppressed
- Service Health = Degraded if the watched attribute value is Major
- Service Health = Slightly Degraded if the watched attribute value is Minor
- Service Health = Up if the watched attribute value is Normal

**Condition High Sensitivity (Default Policy)**

- When any 1 resource(s) are Down then the service is Down.
- When any 1 resource(s) are Degraded then the service is Degraded.
- When any 1 resource(s) are Slightly Degraded then the service is Slightly Degraded.

**Condition Low Sensitivity**

- When all resources are Down then the service is Down.
- When all resources are Degraded then the service is Degraded.
- When all resources are Slightly Degraded then the service is Slightly Degraded.
- When any 1 resource(s) are Down then the service is Degraded.
- When any 1 resource(s) are Degraded then the service is Slightly Degraded.

**Condition Redundancy**

- When all resources are Down then the service is Down.
- When all resources are Degraded then the service is Degraded.
- When all resources are Slightly Degraded then the service is Slightly Degraded.
- When any 1 resource(s) are Down then the service is Slightly Degraded.

**Condition Percentage**

- When 75 percent of the resources are Down then the service is Down.
- When 50 percent of the resources are Down then the service is Degraded.
- When 25 percent of the resources are Down then the service is Slightly Degraded.

**More information:**

[Policy ID Mappings](#) (see page 183)

## Response Time Policies

Watched attribute: LatestErrorStatus

Default reason: Bad Response Time

**Value Mapping**

- Service Health = Down if the watched attribute value is Timeout or Threshold\_Critical
- Service Health = Degraded if the watched attribute value is Threshold\_Major
- Service Health = Slightly Degraded if the watched attribute value is Threshold\_Minor
- Service Health = Up if the watched attribute value is OK

**Response Time High Sensitivity**

- When any 1 resource(s) are Down then the service is Down.
- When any 1 resource(s) are Degraded then the service is Degraded.

**Response Time Low Sensitivity**

- When all resources are Down then the service is Down.
- When all resources are Degraded then the service is Degraded.
- When any 1 resource(s) are Down then the service is Degraded.
- When any 1 resource(s) are Degraded then the service is Slightly Degraded.

**Response Time Redundancy**

- When all resources are Down then the service is Down.
- When all resources are Degraded then the service is Degraded.
- When any 1 resource(s) are Down then the service is Slightly Degraded.

**Response Time Percentage**

- When 75 percent of the resources are Down then the service is Down.
- When 50 percent of the resources are Down then the service is Degraded.
- When 25 percent of the resources are Down then the service is Slightly Degraded.

**More information:**

[Policy ID Mappings](#) (see page 183)

## Service Health Policies

Watched attribute: RM\_Condition

Default reason: Bad Service Health

**Value Mapping**

- Service Health = Down if the watched attribute value is Down
- Service Health = Degraded if the watched attribute value is Degraded
- Service Health = Slightly Degraded if the watched attribute value is Slightly Degraded
- Service Health = Up if the watched attribute value is Up

#### Service Health High Sensitivity

- When any 1 resource(s) are Down then the service is Down.
- When any 1 resource(s) are Degraded then the service is Degraded.
- When any 1 resource(s) are Slightly Degraded then the service is Slightly Degraded.

**Note:** This is the default Policy for a SM\_Service model that monitors resource monitor models/monitor groups/SM\_AttrMonitor models.

#### Service Health Low Sensitivity

- When all resources are Down then the service is Down.
- When all resources are Degraded then the service is Degraded.
- When all resources are Slightly Degraded then the service is Slightly Degraded.
- When any 1 resource(s) are Down then the service is Degraded.
- When any 1 resource(s) are Degraded then the service is Slightly Degraded.

#### Service Health Redundancy

- When all resources are Down then the service is Down.
- When all resources are Degraded then the service is Degraded.
- When all resources are Slightly Degraded then the service is Slightly Degraded.
- When any 1 resource(s) are Down then the service is Slightly Degraded.

#### Service Health Percentage

- When 75 percent of the resources are Down then the service is Down.
- When 50 percent of the resources are Down then the service is Degraded.
- When 25 percent of the resources are Down then the service is Slightly Degraded.

#### More information:

[Create a Service](#) (see page 51)

[Policy ID Mappings](#) (see page 183)

## Contact Status Policies

Watched attribute: Contact\_Status

Default reason: Bad Contact Status

**Value Mapping**

- Service Health = Down if the watched attribute value is Lost
- Service Health = Up if the watched attribute value is Established

**Contact Status High Sensitivity**

- When any 1 resource(s) are Down then the service is Down.

**Contact Status Low Sensitivity**

- When all resources are Down then the service is Down.
- When any 1 resource(s) are Down then the service is Degraded.

**Contact Status Redundancy**

- When all resources are Down then the service is Down.
- When any 1 resource(s) are Down then the service is Slightly Degraded.

**Contact Status Percentage**

- When 75 percent of the resources are Down then the service is Down.
- When 50 percent of the resources are Down then the service is Degraded.
- When 25 percent of the resources are Down then the service is Slightly Degraded.

**More information:**

[Policy ID Mappings](#) (see page 183)





# Appendix B: Resource Monitor Implementation

---

This section contains the following topics:

[Policy Implementation: Monitor Routers](#) (see page 193)

[Resource Monitor Implementation: Monitor Routers and Their Ports](#) (see page 194)

[Refined Resource Monitor Implementation: Monitor Routers, Ports, and Response Time Tests](#) (see page 195)

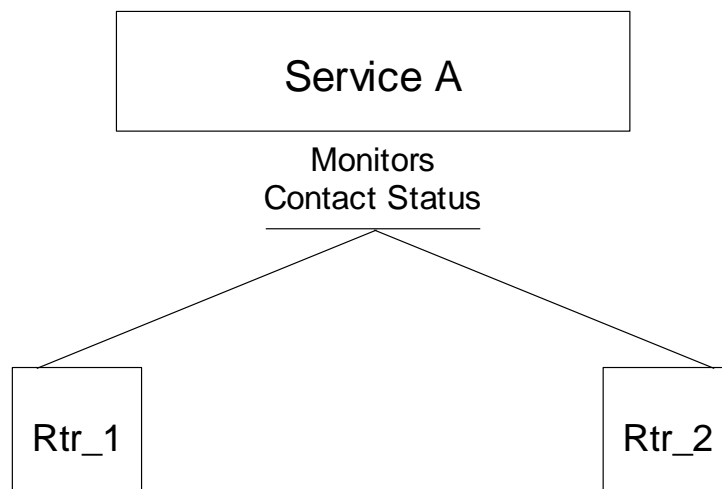
## Policy Implementation: Monitor Routers

Service A monitors two routers: Rtr\_1 and Rtr\_2. Service A functions if Rtr\_1 or Rtr\_2 is up, but it cannot function if both routers are down. Service A uses the Contact Status Low Sensitivity policy to monitor the service.

This policy rule set stipulates the following information:

- If all resources are down, service is down.
- If any 1 resource(s) are down, service is degraded.

### Service with Policy



Service A seems to adequately monitor Rtr\_1 and Rtr\_2, but, on second glance, not enough to determine Service A actual viability. What about the case where Service A is not only dependent on the routers but also on particular router ports? The Contact Status Policy only monitors whether the routers are up or down.

The contact status of the routers could be established, simultaneously the ports on which Service A depends could be unavailable. Because of the limited scope of monitoring provided by its policy, Service A appears viable when actually it is not. Service A requires the more precise method of monitoring its resources that resource monitors provide.

## Resource Monitor Implementation: Monitor Routers and Their Ports

Scrutiny of the policy implementation reveals that it does not monitor the status of router ports that support Service A. The ports must be monitored with contact status of the routers. In addition, ports 4 and 5 on each router must be available for Service A to function optimally and that at least two of the four ports must be available for Service A to function adequately.

The following two resource monitors can be created as resources of Service A:

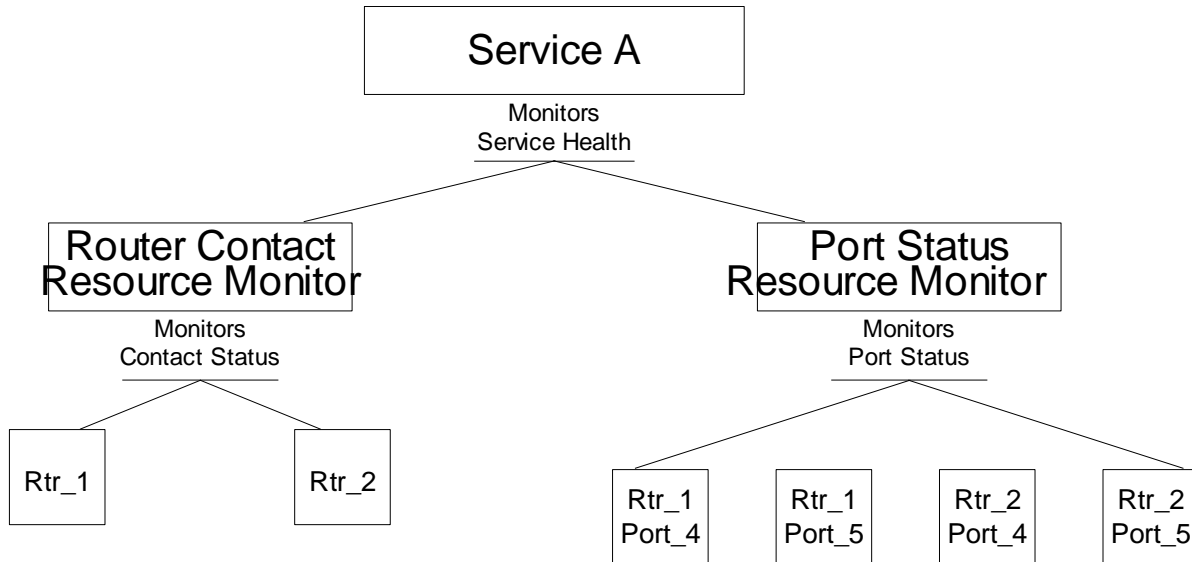
- Router Contact Resource Monitor — Monitors the contact status of Rtr\_1 and Rtr\_2 (same as the policy implementation).
- Port Status Resource Monitor — Monitors the port status of ports 4 and 5 on each router. Because at least two of four ports must be available. This resource monitor uses a Port Status Percentage Policy that reports down when 75 percent of the ports are unavailable and degraded when 50 percent of the ports are unavailable.

The result is that Router Contact Resource Monitor and Port Status Resource Monitor become resources of Service A. Rtr\_1 and Rtr\_2 become resources of the Router Contact Resource Monitor. Ports 4 and 5 on each router become resources of the Port Status Resource Monitor.

Service A monitors the service health attribute of Router Contact Resource Monitor and Port Status Resource Monitor with the Service Health High Sensitivity Policy. This policy rule set stipulates the following information:

- When any 1 resource(s) are Down then the service is Down.
- When any 1 resource(s) are Degraded then the service is Degraded.
- When any 1 resource(s) are Slightly Degraded then the service is Slightly Degraded.

So, if either one of the resource monitors is Down, Service A is down.

**Resource Monitors Monitor Routers and Ports**

## Refined Resource Monitor Implementation: Monitor Routers, Ports, and Response Time Tests

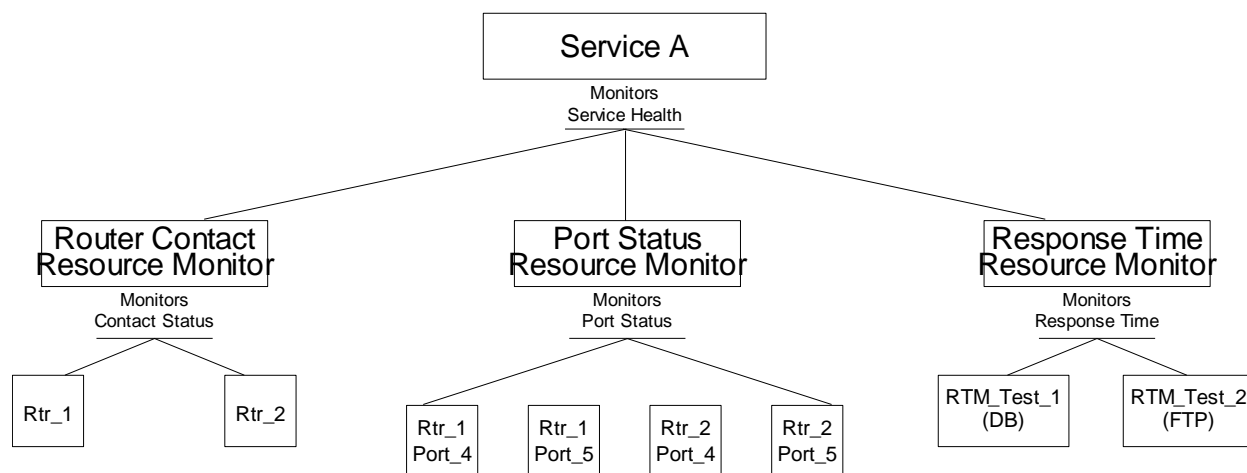
Further scrutiny of service A reveals that database server responsiveness and FTP transfer time are critical to its functionality. Assume that two RTM\_Test models in CA Spectrum define critical, major, and minor thresholds for response time to the database server and the FTP server. These tests can be resources of another resource monitor (the Response Time resource monitor), which becomes another resource of Service A.

Response Time Resource Monitor monitors the response time tests with the Response Time High Sensitivity Policy. This Policy rule set stipulates the following information:

- When any 1 resource(s) are down then the service is down.
- When any 1 resource(s) are degraded then the service is degraded.

Therefore, if this resource monitor reports down (either test indicates unacceptable response time latency) then Service A is down as stipulated by the Service Health High Sensitivity Policy which monitors all Service A Resource Monitors.

**Resource Monitors Monitor Routers, Ports, and Response Time**



# Appendix C: Administration and Maintenance

---

This section contains the following topics:

[Customize a Service Editor Information Table](#) (see page 197)

[Customize a Service Policy Editor Information Table](#) (see page 198)

[Remove Service Manager Historical Data from All Landscapes](#) (see page 198)

[Remove Service Manager Historical Data from a Single Landscape](#) (see page 199)

[Remove Destroyed Service Manager Models from All Landscapes](#) (see page 199)

[Custom Resources Table](#) (see page 200)

## Customize a Service Editor Information Table

You can specify the information tables for services, customers, SLAs, and SLA templates in the Service Editor dialog. Specify the information types (columns) to include, the sort order (by status, by name, or by date for example), and the font and text size. You can revert to default settings. For more information about customizing interface settings, see the *CA Spectrum Operator Guide*.

### Follow these steps:

1. [Open the Service Editor](#) (see page 18).
2. Select the tab for which you want to customize an information table and right-click any column heading.

The Table Preferences dialog appears.

3. Configure the table properties and click OK.

The information table is customized.

## Customize a Service Policy Editor Information Table

You can specify the information tables for policies, attribute maps, and rule sets in the Service Policy Editor dialog. Specify the information types (columns) to include, the sort order (by status, by name, or by date, for example), and the font and text size. You can also revert to default settings. For more information about customizing interface settings, see the *CA Spectrum Operator Guide*.

**Follow these steps:**

1. [Open the Service Editor](#) (see page 18).
2. Select the tab for which you want to customize an information table and right-click any column heading.

The Table Preferences window appears.

3. Configure the table properties and click OK.

The information table is customized.

## Remove Service Manager Historical Data from All Landscapes

You can remove Service Manager historical data from all landscapes.

**Note:** Once you remove historical data from the database, you can no longer generate reports about them.

To remove Service Manager historical data from all landscapes, run the following script:

- Windows:  
`<$SPECROOT>\bin\SMIntializedDB.bat`
- UNIX/Linux:  
`<$SPECROOT>/bin/SMIntializedDB`

## Remove Service Manager Historical Data from a Single Landscape

You can remove Service Manager historical data from a single landscape.

**Note:** Once you remove historical data from the database, you can no longer generate reports about them.

To remove Service Manager historical data from a single landscape, run the following script:

- Windows:

```
<$SPECROOT>\bin\SMInitializeLandscape.bat Live Health
```

**lh**

Indicates landscape handle.

- UNIX/Linux:

```
<$SPECROOT>/bin/SMInitializeLandscape Live Health
```

**lh**

Indicates landscape handle.

## Remove Destroyed Service Manager Models from All Landscapes

You can remove destroyed Service Manager models from all landscapes.

**Note:** Once you remove destroyed Service Manager models from the database, you can no longer generate reports about them.

To remove destroyed Service Manager models (services, SLAs, and so on) and historical data for those models from all landscapes, run the following script:

- Windows:

```
<$SPECROOT>\bin\SMRemoveDestroyedModels.bat
```

- UNIX/Linux:

```
<$SPECROOT>/bin/SMRemoveDestroyedModels
```

## Custom Resources Table

If a service using a policy with a custom attribute map is created, customize the Resources table to display the data you want to view about the monitored attribute that is specified in the custom attribute map. The Resources table appears in the following locations:

- Resources link under the OneClick Information tab
- List tab in OneClick Contents panel
- Service Dashboard List tab
- Resources tab in Service Editor

For more information about customizing OneClick interface elements, see the *OneClick Customization Guide*.

The following table lists Resources table configuration files for attributes that are monitored by standard Service Manager attribute maps:

Attribute	Attribute ID	File
Contact Status	0x10004	table-resources-0x10044-config.xml
Condition	0x1000a	table-resources-0x1000a-config.xml
ConditionValue	0x1000b	table-resources-0x1000b-config.xml
Port Status	0x10f1b	table-resources-0x10f1b-config.xml
Service Health	0x12a40	table-resources-0x12a40-config.xml
Response Time	0x456008c	table-resources-0x456008c-config.xml

These default files are located in the following directory:

```
<$SPECROOT>/tomcat/webapps/CA Spectrum/WEB-INF/slm/config
```

Suppose, for example, you have a service that monitors “load in” data on a set of port resources. In this case, you can create a custom Resources table to display NRM\_PortLoadIn (0x12aad) attribute data for a set of ports that are monitored by the service instead of the default Condition attribute data. The NRM\_PortLoadIn (0x12aad) attribute example is used in the sections that describe how to configure the custom file.



## Create the Custom Table File

You can create the custom file from scratch and can save it to the custom file directory, or you can save a modified version of the default table-resources-config.xml file from the following directory to the custom file directory:

```
<$SPECROOT>/tomcat/webapps/spectrum/WEB-INF/slm/config
```

**Follow these steps:**

1. Create the following custom file directory:  

```
<$SPECROOT>/custom/slm/config
```
2. Save all custom Resources table configuration files for Service Manager to this directory.
3. Create the custom file using the following naming convention:

```
table-resources-<attribute ID>-config.xml
```

For the load in attribute example:

```
table-resources-0x12aad-config.xml
```

## Example: Resources Table Configuration File

The following script shows an abbreviated example of the table-resources-0x12aad-config.xml configuration file. OneClick loads the table that is specified by this file to display the “load in” data column with other types of data for the port resources that are monitored by the service.

Elements pertaining to the “load in” example are highlighted in bold.

```
<?xml version="1.0" encoding="utf-8"?>

<table id="table-resources-0x12aad-config"
      xmlns="http://www.aprisma.com"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.aprisma.com
                          ../../common/schema/table-config.xsd">

  <orientation>horizontal</orientation>

  <swing-table-template>
    <show-vertical-lines>true</show-vertical-lines>
    <show-horizontal-lines>false</show-horizontal-lines>
  </swing-table-template>

  <swing-header-row-template>
    <static-color idref="row-header-color-config"/>
  </swing-header-row-template>

  <swing-row-template>
    <enumerated-color idref="alternatingrow-color-config"/>
  </swing-row-template>

  <column-list>
    <column>
      <name>Load In</name>
      <content>
        <attribute>0x12aad</attribute>
      </content>
      <default-width>125</default-width>
    </column>

    <column idref="column-normalizedstatus-config">
      <default-width>125</default-width>
    </column>

    <column idref="column-modelname-config">
      <default-width>125</default-width>
    </column>
  </column-list>

  <default-sort>
    <sort-column-list>
      <sort-column>
        <name>
          com.aprisma.spectrum.app.topo.client.interfaces.render.NormalizedStatusColumn
        </name>
```

```
        <direction>ascending</direction>
    </sort-column>
    <sort-column>
        <name>
            com.aprisma.spectrum.app.util.render.ModelNameColumn
        </name>
        <direction>ascending</direction>
    </sort-column>
</sort-column-list>
</default-sort>
</table>
```

**More information:**

[Create the Custom Table File](#) (see page 201)



# Index

---

## A

- AdministratorRW • 16
- alarm exemption list • 133
- alarm types
  - adding to a custom condition policy • 78
  - and service health • 57, 77
- alarms
  - and service health • 77
- associating
  - customers with services • 65
  - customers with SLAs • 113
  - owners with services • 65
  - SLAs to a service • 135
- attribute map properties • 82
- attribute maps
  - about • 12, 75
  - attcreating • 82, 84
  - custom resources table • 200
  - deleting • 85
  - editing • 84
- attributes
  - Condition • 57, 183
  - Condition\_Value • 183, 185, 187
  - Contact\_Status • 123, 183, 190
  - LatestErrorStatus • 183, 188
  - MonitorPolicy\_ID • 122
  - Port\_Status • 183, 186
  - RM\_Condition • 183, 189
  - Schedule • 147
  - SM\_AttrMonitor • 124, 142, 189
  - SM\_Customer • 143
  - SM\_CustomerGroup • 143
  - SM\_Guarantee • 145
  - SM\_Service • 141
  - SM\_SLA • 144
- availability guarantees • 105, 106, 107
- available schedules list • 64

## C

- Condition • 183
- Condition High Sensitivity • 51, 187, 189
- Condition Low Sensitivity • 187
- Condition Percentage • 187
- condition policies • 187

- Condition Redundancy • 187
- Condition\_Value • 183, 185, 187
- Contact Status High Sensitivity • 190
- Contact Status Low Sensitivity • 190
- Contact Status Percentage • 190
- Contact Status Redundancy • 190
- Contact\_Status • 123, 183, 190
- creating
  - an SLA from an SLA template • 103
  - attribute maps • 82, 84
  - custom table files • 201
  - customer groups • 93
  - customers • 92
  - guarantee templates • 116
  - guarantees • 106, 107, 135
  - maintenance schedules • 63
  - policies • 77, 79
  - resource monitors • 56
  - rule sets • 87, 88
  - services • 51
  - SLA periods • 111
  - SLA templates • 114
  - SLAs • 101
- current schedules list • 64
- custom condition policies • 78
- custom file directory • 201
- custom resources table • 200
- custom table file • 201
- customer groups
  - about • 91
  - attributes • 143
  - creating • 93
  - deleting • 95
  - editing • 94
  - moving • 94
- customer reports • 178
- CustomerManager tag • 120, 137
- customers
  - about • 14, 91
  - associating with a service • 65
  - associating with an SLA • 113
  - attributes • 143
  - creating • 92
  - criticality value • 92
  - deleting • 95

---

- editing • 93
- moving • 94

## D

dashboard views • 160

defining

- alarm exception list • 133
- customer groups • 137
- customers • 137
- service maintenance schedules • 133
- SLAs • 136

deleting

- attribute maps • 85
- customer groups • 95
- customers • 95
- guarantee templates • 118
- guarantees • 111
- policies • 80
- rule sets • 89
- services • 62
- SLA templates • 116
- SLAs • 113

destroyed models • 199

detailed availability reports • 177

Distributed SpectroSERVER environment • 20

## E

editing

- attribute maps • 84
- customer groups • 94
- customers • 93
- guarantee templates • 117
- guarantees • 110
- policies • 79
- rule sets • 88
- services • 61
- SLA templates • 115
- SLAs • 112
- topology views • 160

examples

- associating an SLA to a service • 135
- creating guarantees • 135
- defining an alarm exemption list • 133
- defining customer groups • 137
- defining customers • 137
- defining service maintenance schedules • 133
- defining SLAs • 136
- global collections • 69, 71

- importing XML input files • 140
- multiple global collections • 73
- resources table configuration file • 201
- service models • 67, 68
- services • 67, 123, 124

## G

Generate\_Service\_Alarms • 123

global collections • 69, 71, 73

guarantee properties

- accumulation method • 107
- control • 107
- generate warning alarm • 107
- guarantee name • 107
- guarantee type • 107
- outage type • 107
- violation threshold • 107

guarantee templates

- about • 116
- creating • 116
- deleting • 118
- editing • 117

guarantee types • 105

guarantees

- about • 98
- and specifying business hours • 109
- and threshold types • 106
- attributes • 145
- creating • 107
- deleting • 111
- editing • 110
- types of • 105

## H

health reports • 181

historical data • 198, 199

## I

inventory reports • 176

## L

LatestErrorStatus • 183, 188

licenses • 16

list views • 153

Locator tab • 20

---

## M

maintenance schedules • 63, 64, 133

Modeling Gateway

creating models with • 121, 143

importing configuration files with • 140

modeling services with • 122

monitor routers • 193, 194, 195

MonitorPolicy\_ID • 122

moving

customer groups • 94

customers • 94

## O

OneClick

administrator role • 16

Console • 171

licenses • 16

operator role • 16

OperatorRW • 16

outage history • 155, 163

outage reports • 175

outages • 162

owner reports • 173

## P

periods • 99

policies

about • 12, 75, 122

creating • 77, 79

deleting • 80

editing • 79

port status • 186

response time • 188

service health • 189

statistical • 76

status • 76

policyrep.jsp • 122, 183

Port Status High Sensitivity • 186

Port Status Low Sensitivity • 186

Port Status Percentage • 186

Port Status Redundancy • 186

Port\_Status • 183, 186

printing • 159

## R

reports

detailed availability type • 177

generating • 182

health type • 181

inventory type • 176

outage type • 175

owner type • 173

SLA status type • 179

summarized availability • 177

resource monitoring • 56

resource monitors • 12, 25, 54

resources table configuration file • 201

Response Time guarantees • 105, 106, 107

Response Time High Sensitivity • 188

Response Time Low Sensitivity • 188

Response Time Percentage • 188

Response Time Redundancy • 188

RM\_Condition • 183, 189

roll-up indications • 22

root cause alarms • 92

router ports • 194

rule sets

about • 12, 75, 85

creating • 87, 88

deleting • 89

editing • 88

## S

Schedule • 147

searching

for customers in OneClick • 20

for services in OneClick • 20

for SLAs in OneClick • 20

Service Dashboard

about • 22, 149, 157

and current outages • 155

and outage history • 155

and service outages • 163

component details panel • 151

contents panel • 151

editing tools • 160

navigation panel • 151

printing views • 159

viewing • 151, 171

Service Editor

about • 18

information table • 197

opening • 18

service health • 57

Service Health High Sensitivity • 189

---

- Service Health Low Sensitivity • 189
- Service Health Percentage • 189
- Service Health Redundancy • 189
- service health values • 14
- Service Level Manager portlet • 11, 167, 168, 169, 170, 171
- service management
  - components • 155, 158
- Service Manager
  - about • 11, 15
  - licensing • 151
- service models • 121
- service outages • 163
- Service Policy Editor
  - about • 18, 19
  - information table • 198
  - opening • 19
- service properties
  - Containers • 51
  - Criticality • 51
  - Description • 51
  - Generate Service Alarms • 51
  - In Maintenance • 51
  - Landscape • 51
  - Name • 51
  - Security String • 51
- ServiceManagerRW • 16
- services
  - about • 12, 25
  - adding to a resource • 60
  - associating with customers • 95
  - associating with owners • 65
  - configuring • 123
  - creating • 51
  - deleting • 62
  - editing • 61
- SLA Manager • 136
- SLA period • 155
  - creating • 111
- SLA properties
  - Control • 101
  - Description • 101
  - Expiration Date • 101
  - Notes • 101
  - Period • 101
  - Security String • 101
  - SLA Name • 101
- SLA status reports • 179
- SLA templates

- about • 114
- creating • 114
- deleting • 116
- editing • 115

SLAs

- about • 14, 97
- associating with customers • 95
- attributes • 144
- creating • 101, 103
- defining • 136
- deleting • 113
- editing • 112

SM\_AttrMonitor • 124, 142, 189

SM\_Customer • 143

SM\_CustomerGroup • 143

SM\_Guarantee • 145

SM\_Service • 141

SM\_Service\_Mgt • 120

SM\_ServiceMgr • 120

SM\_SLA • 144

SM\_SLA\_Mgr • 120

statistical policies • 76

status indicators • 154

status policies • 76

summarized availability reports • 177

## T

- topology views • 153, 160
- trending
  - about • 155
  - violation threshold • 155
  - warning threshold • 155

## U

- Unicenter Management Portal (UMP) • 11, 22, 167, 168
- utilities • 18

## V

- Value Mapping • 188, 190
- viewing
  - current outages • 162
  - customer information • 170
  - outage history • 163
  - service information • 169
  - SLA information • 170



---

## X

XML framework • 120

XML input files • 120, 140