

CA Spectrum®

Condition Correlation 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 guide references CA Spectrum®.

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.

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: Introduction	9
About Condition Correlation	9
Condition Correlation Components	9
The Condition Correlation Editor	12
Open the Condition Correlation Editor	13
Condition Correlation Import and Export Features	14
How to Create a Condition Correlation Domain	14
 Chapter 2: Creating and Managing Conditions	 17
Create a Condition	17
Create a Parameter	18
Manage a Parameter	19
Manage a Condition	20
 Chapter 3: Creating and Managing Rules	 23
Create a Rule	23
Manage a Rule	26
Topology Information	26
Update Topology Operators in Configuration Files	27
 Chapter 4: Creating and Managing Policies	 29
Create a Policy	29
Manage a Policy	29
 Chapter 5: Creating and Managing Domains	 31
About Correlation Domains	31
Create a Domain in the Condition Correlation Editor	32
Create a Domain in the OneClick Console	33
Manage a Domain	33
 Chapter 6: Testing and Debugging	 35
How to Develop and Test Correlations	35
Guidelines and Best Practices	36
Testing a Correlation	36

Test the Correlation with the Command-Line Interface	37
Test the Correlation with the Web Services API	38
Verify the Simulated Events	39
Chapter 7: Debugging Correlations	41
Debugging Prerequisites	41
Debugging Tools	41
Appendix A: Condition Correlation Examples	45
How to Configure a Condition Correlation for a Power Outage	46
Create Power_Outage and Battery_On Conditions	47
Create Rules to Define the Symptoms of the Power Outage Alarm	48
Create a Power_Outage Policy	50
Create a Backup_Power Domain and Add Resources	50
Verify the Correlation	51
Disk Full Scenario	52
EventDisp Entries	52
How to Configure the Sample DiskFull Condition Correlation	54
Create a Clear Events Correlation	61
Create an Additional Parameter for the DiskFull Condition	61
Create an Event Rule to Identify a Cleared Disk Problem Alarm	62
Log and Add an Event to Clear the DiskFull Alarms	62
Create the Conditions Required for the Clear Correlation	63
Create a Rule to Clear DiskFull Alarms	64
WAN Link Failure Example	64
WAN Link Scenario	65
WAN Link Correlation Strategy	65
WAN Link Failure Configurations	66
Appendix B: Special Topics	67
Condition Correlation and Fault Isolation	67
About Transfer Rules	67
Advanced Correlations and Data Type Comparisons	68
Appendix C: REST Examples for Correlation Testing	69
RESTful Web Services XML Example – No Event Variables	69
RESTful Web Services XML Example – with Event Variables	70
Configure WizTools RESTClient	71
Create and View Simulated Alarms: An Example	72

Chapter 1: Introduction

This section contains the following topics:

[About Condition Correlation](#) (see page 9)

[The Condition Correlation Editor](#) (see page 12)

[How to Create a Condition Correlation Domain](#) (see page 14)

About Condition Correlation

CA Spectrum Condition Correlation supports events, troubleshooting, and root-cause analysis. The Condition Correlation component lets you set up a system in CA Spectrum to determine the root-cause alarm from a heterogeneous group of managed infrastructure resources (models). You can use Condition Correlation to select the criteria that identify a causal problem event. Such events precipitate a specific set of events, which are in turn identified as symptoms. You can select a set of resources (models) for the correlation to consider and define it as the correlation domain.

Condition Correlation provides the following benefits:

- Respond to the real problem efficiently. Spend less time responding to symptomatic problems.
- Track problem trends and interdependencies.
- Respond quickly to changes in the infrastructure. You can manage multiple Condition Correlation implementations from a single landscape.

Condition Correlation Components

You can use Condition Correlation to construct a system of components that define fault indicators. You can use these components to create a process for fault association. Fault indicators specify the resources that are evaluated by the system. The following components are available to you:

- [Conditions](#) (see page 10)
- [Rules](#) (see page 10)
- [Policies](#) (see page 12)
- [Correlation Domains](#) (see page 12)

Before you begin configuring Condition Correlation, we recommend reviewing the predefined component settings. You can see these in the Condition Correlation Editor.

More information:

[Open the Condition Correlation Editor](#) (see page 13)

Conditions

Conditions are fundamental building blocks of the correlation system. A condition, like a CA Spectrum alarm, is a transitory occurrence on a resource, such as status change. A condition exists as long as the criteria that produced the condition are met. As with an alarm, a *set* event always initiates a condition, and a *clear* event clears a condition. When you define a condition, you identify the set and clear event types.

A 'set' event creates an alarm that is associated with the condition. Therefore, a condition can be cleared when the associated alarm is destroyed. Similarly, a condition is also cleared when a rule creates the condition (through its 'set' event), and no set of conditions still fulfills the rule. In this case, the condition is cleared automatically. A condition that a rule created through its set event is an *implied condition*.

You can define the conditions that correspond to CA Spectrum alarms. If the 'set' event of the condition is same as the set event of the alarm, the condition instance is instantiated after the alarm is generated. The alarm itself is linked within the correlation system. This link lets Condition Correlation hide symptomatic alarms from the main alarm list in OneClick and relate symptomatic alarms to root-cause alarms. The symptomatic alarms are listed in the Symptoms list of the root-cause alarm under the Impact tab.

Important! Alarms that are available at startup are not correlated.

You can also define the parameters for a condition that are used to establish correlation criteria when you create correlation rules. A *parameter* can be any event variable data or any model attribute of the model that is associated with the condition. You can create new parameters, or you can create modified versions of existing parameters.

Note: A correlation condition has no relationship with the condition attribute for a CA Spectrum model.

Rules

A *rule* defines the relationship between two or more conditions when specific criteria are met. You can define a rule to stipulate that one condition is a symptom of, or the cause of, another.

For example, you can associate a symptomatic SPM test threshold violation condition to a root-cause port LinkDown condition. You can apply this rule in a policy, to a set of SPM test and port models in a domain. In addition, you can create a rule to indicate that one or more conditions imply that another exists.

Rule Patterns

You can express rules in any of the following patterns:

Caused By

Condition Z causes Condition X or a set of conditions.

A correlation is made when all of the symptom conditions exist, the rule criteria apply, and the root cause condition Z exists. If Z is associated with an alarm, all symptomatic alarms are hidden under that alarm. The condition (color) of the model remains the same as before. For example, if one yellow alarm on the model hides another red alarm on the model, the other model remains red with no alarms displayed.

Note: When any of the conditions are cleared, the correlation is not broken.

Implies

Condition X or a set of conditions implies Condition Z.

When all of the symptoms exist and the rule criteria apply, the root cause Condition Z is created. A 'set' event is therefore created for Condition Z, which can then create an alarm. Condition Z is only cleared if any of the symptoms are subsequently cleared, and if no other set of conditions still supports the rule. But if the condition creates an alarm, the alarm is only cleared if the condition has a 'clear' event, which must clear the alarm. Therefore, the alarm can remain, depending on its configuration.

Implied Cause

Condition X or a set of conditions is the implied cause of Condition Z.

The Caused By and Implies pattern combines both of the previous patterns.

Important! Correlation using the same condition as the symptom and the cause fails.

You cannot set up a correlation using the same condition (such as implied cause) as both symptom and cause. However, you can create another condition with the same set or clear events and can use the condition as the root cause.

Example

Set up Condition A, and Rule A implies Caused by A on the correlation domain. When Alarm A is created on a device in the domain, you can see that another Alarm A is created on the correlation domain model. However, the Alarm A on the device does not become a symptom of the domain alarm.

To make the alarm a symptom of the domain alarm, you can create a B condition, similar to A, with a rule that Condition A implies Caused by B on the correlation domain.

Other Patterns

Condition Correlation lets you construct more granular rule patterns using more rule criteria that must be met before a correlation is established between two conditions. You can specify the criteria by comparing the parameters of one condition with another or in terms of specific values.

For example, an instance of a LinkDown condition on a port model can be caused by an instance of a BoardPulled condition on a board. This relationship can occur if the slot number of the port is equal to the slot number of the board, and both the port and the board are from the same device.

Policies

A *policy* is a set of one or more rules. You can group any number of rules in a policy. You can apply one or more policies to any number of resource groups (in a domain).

Use policies to simplify the implementation of rules for multiple domains. All implementations of a policy are updated after you add, edit, or remove rules from a policy.

Correlation Domains

A *correlation domain* is a group of resources that is created as a CA Spectrum container model. Condition Correlation assesses these resources collectively. This assessment is based on the rules in the policies that are applied to it. A domain can include any number of models of various model types and can have any number of policies applied. Therefore, when you select the resources in a domain, you are also deciding what is evaluated by the policy or policies that are applied to it.

You have multiple options for creating a correlation domain and populating it with resources. You can add resources on a per-resource basis. Or you can create a domain from a service or Global Collection model, which are entities that represent collections of resources.

The Condition Correlation Editor

The Condition Correlation Editor window lets you create and manage correlation system components. The window also lists and provides access to all predefined and custom (user-defined) components.

The Condition Correlation Editor window contains the following tabs:

Conditions

Lists the predefined and custom conditions. Select a condition to see a list of corresponding parameters.

Note: Not all conditions include parameters.

Rules

Lists the predefined and custom rules. When you select a rule, the corresponding correlation criteria of that rule are listed in the Rule Criteria tab.

Policies

Lists the predefined and custom policies. Select a policy to see the corresponding rules of that policy on the Rules tab.

Domains

Lists the predefined and custom domains. When you select a domain, the corresponding policies of that domain and resources are listed in the Policies tab and the Resources tab respectively.

The Condition Correlation Editor provides buttons to let you create, edit, copy, and delete conditions, rules, policies, or domains.

Use the Filter field to specify the condition, rule, policy, or domain entries to display in the editor window.

Open the Condition Correlation Editor

The Condition Correlation Editor lets you configure all Condition Correlation component settings. You must have OneClick administrative privileges to access the Condition Correlation Editor.

Follow these steps:

1. Log in to OneClick.
2. Select Tools, Utilities, and then Condition Correlation Editor.

The Condition Correlation Editor window opens. By default, it displays the Conditions tab list and any parameters that are defined for the selected condition.

Condition Correlation Import and Export Features

You can import or export correlation data using the Import or Export features in the Condition Correlation editor. The following options are available in the Condition Correlation editor:

Export

Exports the correlation data and saves it to an XML file.

Import

Imports the correlation data from an XML file. The three scenarios that can occur while you import the data are as follows:

SKIP

Defines the entry that already exists. The skipped entries are prefixed with [SKIP].

REPLACE

Replaces the existing entry of the same name and type only if you enable the Replace Existing option. The replaced entries are prefixed with [REPLACE].

IMPORT

Defines all other entries that are not prefixed with SKIP or REPLACE. These entries are prefixed with [IMPORT].

Note: The import and export of domains is not supported.

How to Create a Condition Correlation Domain

Deploying a correlation system to a particular group of managed infrastructure resources is synonymous with creating a correlation domain. Once the domain is created on a landscape, the correlation system is in effect.

Note: Condition correlations are implemented in a SpectroSERVER or in multiple SpectroSERVERs. Therefore, condition correlations are not affected if you start or stop the OneClick web server.

Verify the following information before configuring the required domain parameters:

- A domain has at least one policy that is applied to it.
- The policy includes at least one rule.
- The rule criteria are logically appropriate for the conditions it evaluates for a correlative association.

Important! Attempt to produce a problem to manage and test the correlation system before you deploy the system in a production environment.

Perform the following tasks to create a condition correlation domain:

1. Create a domain and add the resources that you want include in it. In a later step, you can apply one or more correlation policies to the domain.

Once you have created a domain, you can add resources to it and can remove resources from it at any time.

2. Create one or more conditions that you want to be evaluated by correlation rule criteria.

Note: If you want to use available conditions, skip this step.

3. Create the rule or rules that establish root-cause condition and symptomatic condition associations if criteria specified by the rules are met.

Note: If you want to use available rules (such as rules that specify predefined conditions), skip this step.

A rule evaluates two or more conditions. If rule criteria are met, Condition Correlation identifies one condition as the root-cause condition and the other conditions as symptomatic of the root-cause condition. Once you have created a rule, you can modify its criteria or the conditions it evaluates at any time.

4. Create the policy or policies that contain the correlation rules to associate with the domain.

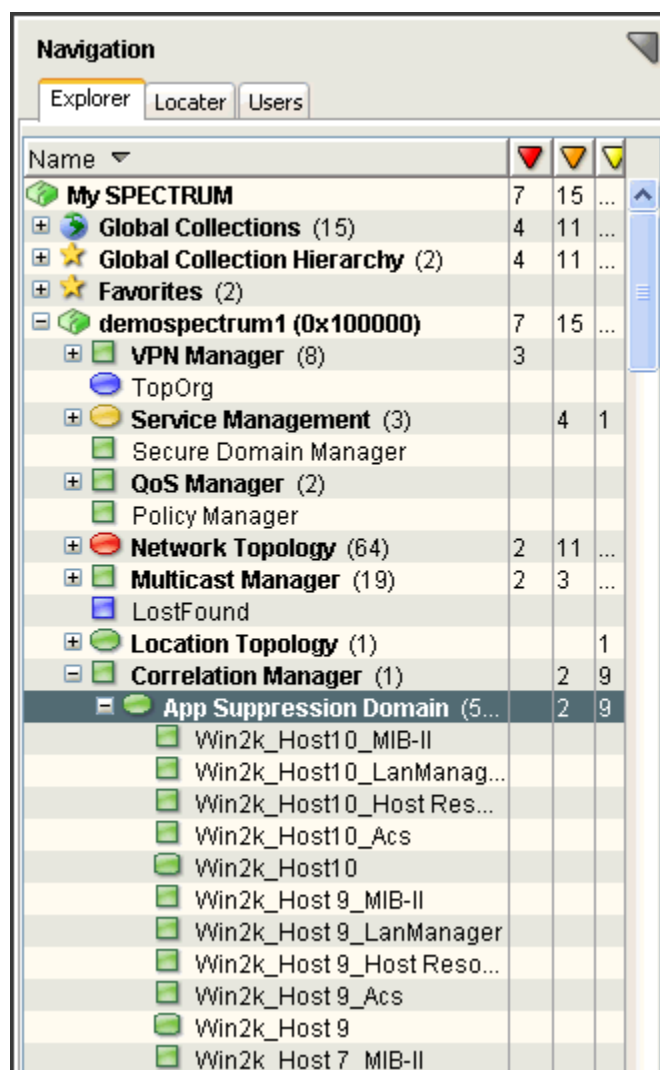
Note: If you want to use available policies, skip this step. You can add rules to or can remove rules from a policy at any time.

5. Apply one or more policies to the domain.

Note: Existing correlation domains adjust to policy changes automatically, keeping the correct correlation state.

The Condition Correlation process is in effect for the resources that are included in the domain. The domain is modeled as a correlation domain container in OneClick.

The following image is an example for domain container and the resources included in it:



Chapter 2: Creating and Managing Conditions

This section contains the following topics:

[Create a Condition](#) (see page 17)

[Manage a Condition](#) (see page 20)

Create a Condition

Conditions are fundamental building blocks of the correlation system. A Condition, like a CA Spectrum alarm, is a transitory occurrence on a resource, such as status change. A condition exists as long as the criteria that produced the condition are met. As with an alarm, a set event always initiates a condition, and a clear event clears a condition.

Follow these steps:

1. [Open Condition Correlation Editor](#) (see page 13).

The Condition Correlation Editor window opens.

2. Click the Conditions tab.

A list of conditions is displayed.

3. Click  (Create).

The Create Correlation Condition dialog opens.


4. Specify a value for the following condition properties:

Condition Name

Defines the condition. For example, supply the names Power_Outage and Battery_On.

Set Event Code

Identifies the CA Spectrum event code that is associated with the condition. When you define a condition, the set and clear event types are identified

5. (Optional, for an advanced correlation only) Click  (Create) in the Parameters section to specify parameters.

The Create Parameter dialog opens. You can create parameters for the conditions as desired. For more information, see [Create Parameter](#) (see page 18).

6. Click OK.

A new condition is created and added to the Conditions tab list. The Author property identifies you as the condition author.


Create a Parameter

A condition *parameter* can be any event variable data or any model attribute of the model that is associated with the condition.

Parameter values are filled in at the time the condition is created, from the event that created it, or from the model where the event was created. These parameters are then available in the Advanced Rule Criteria section.

For count conditions, a *count* parameter is available automatically, after you select that condition type in the rule.

Follow these steps:

1. Click  (Create) in the Parameters section.

The Create Parameter dialog opens.

2. Provide a value for the following parameter properties:

Parameter Name

Identifies the parameter. Provide a name that indicates the parameter type.

Parameter Type

Specifies the type of parameter. Choose *one* of the following options:

- Model Attribute: Specifies a model attribute parameter type.
- Var Bind: Specifies a Var Bind parameter type.
- Predefined: Specifies a Model, Model Type, or Device Model.

Parameter ID

Identifies the type of parameter.

If you select Model Attribute, click Attribute to open the Attribute Selector dialog and select the appropriate Model Attribute ID.

If you select Var Bind, enter the Var Bind variable number that is associated with the trap for the model.

If you select Predefined, select *one* of the following attributes from the adjacent Parameter Type drop-down list:

- Model: Enter the Model_Handle associated with the condition (Attribute ID 0x129fa).
- Model Type: Enter the Model_Type_Handle of the model that is associated with the condition (Attribute ID 0x10001).
- Device Model: Enter the Device_Mdl_Handle of the model that is associated with the condition (Attribute ID 0x10069).

Use as discriminator

Designates the parameter as a discriminator. This setting lets you clear only the 'set' events that include parameter values that match the values in the 'clear' event. You can designate multiple parameters as discriminators. When condition parameters are designated as discriminators, the condition maintains the parameter values that were in place when the set event produced the condition. A condition can only be cleared if the 'clear' event contains parameter values that match the values in the 'set' event.

To use different discriminators for special situations, you can use the same condition discriminators that the associated alarm uses. If you use the same condition discriminators as the alarm, the conditions match the alarms and clear accordingly.

3. Click OK.

The parameter is created.


Manage a Parameter

You can edit, copy, and delete the parameter values of all parameters that are listed in the Parameters section.

Follow these steps:

1. Select the parameter in the Parameters section and click  (Edit).


The Edit Parameter dialog opens.

2. To copy a parameter, click  (Copy).

The Copy Parameter dialog displays the property conditions of the parameter you selected.

Note: The Parameter Name is suffixed with _COPY because the new parameter is copied from an existing parameter and contains a unique name. If the name is already in use, a Name already exists message appears.

3. Edit the properties of the parameter as necessary, and click OK.

4. To delete a parameter, click  (Delete).

The selected parameters are removed from the Condition Parameters list.

Note: You cannot delete a parameter that is still in use by a rule.

Manage a Condition


In the Condition Correlation Editor window, you can edit, copy, and delete a condition from the list of predefined (CA-authored) and custom (user-authored) conditions. You can permanently delete user-authored conditions, but you cannot delete predefined conditions. If you or another user has edited and assumed ownership of a predefined condition, you can delete it temporarily. The Condition Correlation Editor restores the predefined condition with its default settings when you restart the OneClick server. You cannot delete a condition that is in use by a rule.

Important! Any changes that you make to existing conditions forces Condition Correlation to drop all current conditions of the same type.

Follow these steps:

1. Click the Conditions tab in the Condition Correlation Editor window.

A list of conditions is displayed.

2. Select the condition to edit and click  (Edit).

The Edit Condition dialog opens displaying the property settings of the condition you selected.

3. Edit the values for the following condition properties:

Set Event Code


Identifies the CA Spectrum event code that is associated with the condition.

Clear Event Code

(Optional) Identifies the CA Spectrum clear event code that is associated with the condition.


4. (Optional, for an advanced correlation only) Specify parameters for the selected condition.

Update one or more parameters that can be used to determine a correlation that is made between instances of the specified condition.

5. To copy a condition, click  (Copy).

The Copy Condition dialog opens, displaying the property settings for the condition you selected.

Note: The Condition Name is suffixed with _COPY because the new condition is copied from an existing condition and contains a unique name. If the name is already in use, a "Name already exists" message appears.

6. To delete a condition, click  (Delete).

Condition Correlation removes the conditions from the Conditions tab.

Chapter 3: Creating and Managing Rules

This section contains the following topics:

[Create a Rule](#) (see page 23)

[Manage a Rule](#) (see page 26)

[Topology Information](#) (see page 26)

Create a Rule

A rule defines the relationship between two or more conditions when specific criteria are met. You can define a rule to stipulate that one condition is a symptom of, or the cause of, another condition. For example, you can associate a symptomatic condition with a root-cause condition. You can apply this rule in a policy or to a set of models in a domain. In addition, you can create a rule to indicate that one or more conditions imply that another condition exists.

Follow these steps:

1. [Open the Condition Correlation Editor](#) (see page 13).

The Conditions tab is displayed by default.

2. Click the Rules tab.

A list of rules is displayed.

3. Click  (Create).

The Create Rule dialog opens.

4. Enter a name for the rule in the Rule Name field.

5. (Optional) Click set in the Type column of each item you select, specify the symptom condition to belong to a correlation domain, and select one of the following options:

- **Exists:** The condition is in the correlation domain.
- **Not Exists:** The condition is not present in the correlation domain. This option lets you create rules that can only be satisfied if the condition does not exist in the correlation domain.
- **Counts:** The condition is in the correlation domain, and it enables totals/limits/range comparisons using the Advanced Rule Criteria section of the Create Correlation Rule dialog. This option lets you create rules only if a particular condition exists, reaches a limit, or is in a user-defined range.

When using a condition for counting, a new parameter is automatically created for that condition named "Condition Count." This count can be used in the Advanced Rule Criteria section, as shown in the following example:

TestCondition.Condition Count GREATER_THAN 10.

No other parameter can be used for counted conditions. Because multiple copies are present, Condition Correlation cannot determine the condition from which to derive the parameter value.

6. Select one or more symptom conditions in the Symptom Condition(s) list.

Note: The rule is created based on the selected symptom conditions.

7. Select *one* of the following values from the Relationship drop-down list to specify the relationship between symptomatic conditions and the root cause condition.

- **Caused By:** The alarm that is associated with the root cause condition caused the associated symptomatic conditions. When the rule is met, OneClick suppresses the alarms for symptomatic conditions and lists them as symptoms under the Alarms view Impact tab in OneClick.
- **Implies:** The symptomatic conditions suggest the existence of another condition that can be unknown to the management system. When the rule is satisfied, the set event of the implied condition is processed on the target model. This condition can raise an alarm on the target model, but OneClick does not suppress the alarms for symptomatic conditions.
- **Implied Cause:** This rule incorporates the logic of both the Caused By and Implies rules. The symptomatic conditions are indicative of another condition. The set event of this implied condition is processed on the target model. If this event raises an alarm on the target model, OneClick suppresses the alarms that are associated with the symptomatic conditions. The suppressed alarms are listed as symptoms of the root cause alarm under the Impact tab in OneClick.

Note: If you select Implies or Implied Cause, the Root Cause Target selection box is displayed on the Correlation Rule dialog. Root Cause Target lets you specify the alarm that can be generated on the correlation domain with which the rule is associated or on one of the symptomatic conditions.

To associate the implied alarm (event) with a model, add the predefined "Model" parameter to the condition that you know is created on the target model. Then select this condition and the "Model" parameter as the root cause target from the Root Cause Target section.

You can imply the condition (event/alarm) on a model where you do not have an alarm and include the model in the correlation. Consider the following examples:

- For a container, select the Model Active condition for that model, and add some rule criteria to identify the correct Model Active condition.
- For a port alarm, add the Device Model parameter to the port condition and add a criterion in the rule that specifies "Model Active.Model EQUAL TO PortCondition.Device Model". The implied condition is created on the desired model. The "Model Active" condition is created once for each model participating in the correlation domain.

Note: If you select Implies or Implied Cause, the Clear Symptom condition if Implied Condition is cleared check box is displayed.

8. Select a condition from the Root Cause Condition dialog that caused, or was the implied cause, of the symptomatic conditions.

Note: You can select only one root cause condition for a rule.

9. (Optional) Select the Clear Symptom condition if Implied Condition is cleared check box.

The symptom conditions are cleared when the implied condition is cleared. This feature works with a chain of implications similar to the following scenarios:

- ConditionA implies ConditionB, and ConditionB implies ConditionC
- You assert ConditionA, and the SpectroSERVER then asserts ConditionB and ConditionC
- You clear ConditionC, and the SpectroSERVER then clears ConditionB and ConditionA

10. (Optional) Click Show Advanced.

The Advanced Rule Criteria workspace opens. You can use advanced rule criteria when you have specified condition parameters and you want to establish correlation criteria that are based on parameter or topology values. In addition to the parameter comparison, you can also include topology information (association between models).

11. Click OK.

A new rule is created and added to the Rules tab list. The Author property identifies you as the author of the rule.

More information:

[Update Topology Operators in Configuration Files](#) (see page 27)
[Topology Information](#) (see page 26)


Manage a Rule

In the Condition Correlation Editor, you can edit, copy, and delete rules from the list of predefined and custom rules.

Follow these steps:


1. Click the Rules tab in the Condition Correlation Editor window.

A list of rules is displayed.

2. Select the rule that you want to edit and click  (Edit).


The Edit Rule dialog opens.

Note: You cannot edit a rule name when the rule is specified in a policy.

3. To copy a rule, click  (Copy).

The Copy Rule dialog displays the property settings for the rule.

Note: The suffix _COPY is appended to the Rule Name to provide a unique name for the new rule. A unique name is required.

4. To delete a rule, click  (Delete).

Condition Correlation removes the selected rules from the list on the Rules tab.

5. Edit the properties of the rule as necessary, and click OK.

The Condition Correlation Editor saves your changes.

Topology Information

The Advanced Rule Criteria let you specify topology information when you create a rule. Topology rules create associations between models that are used in the correlation rules. You can insert regular parameter criteria, topology criteria, or both. The topology rule criteria, like parameter criteria, are applied to the parameters of conditions. These condition parameters must be the model handles or must be convertible to model handles.

The operator that is used on condition parameters is a relation procedure. The relation procedure lets the rule verify the existence of the relationship between the two models. The operator stands for the type of relationship.

The Topology operator stands for the following relations:

Regular Relations

Represent regular associations. The rule criteria evaluate to TRUE when the left model is associated with a topology relation to the right model. The following regular relations operators are used:

- Connects_to
- HasPart
- Manages
- Collects
- Correlates

Special Relation

Represents special associations. The rule criteria evaluate to TRUE when the left model is a port of the device model on the right. One special topology relation operator is available: IsPortOf.

Update Topology Operators in Configuration Files

The topology operator is a relation procedure that checks whether the relationship exists between two models. You can use all available topology operators in your correlations. CA Spectrum also lets you update the topology operators in the configuration files.

Follow these steps:

1. Copy the topology association configuration file, `$SPECROOT/tomcat/webapps/spectrum/WEB-INF/event/config/topology-criteria-operator-choices.xml`, to the `$SPECROOT/custom/event/config` area.

Note: Create the `$SPECROOT/custom/event/config` directory if it does not exist.

2. (Optional) Update the configuration file to add, delete, or modify a topology operator.

Note: You can add or remove any number of topology operators from the configuration files.

3. Restart the Tomcat web server.

The updates take effect when you reopen the Condition Correlation Editor.

Example: Add an 'IsAdjacent_to' topology operator

This example adds the IsAdjacent_to topology operator in the configuration file.

```
<criteria-choice>
  <relation-choice>
    <name>IsAdjacent_to</name>
    <verbose>is adjacent to</verbose>
    <relation-id>0x00010007</relation-id>
  </relation-choice>
</criteria-choice>
```

The code has the following parameters:

name

Indicates the name of the relation as defined in CA Spectrum.

verbose

Indicates the verbose text that is shown for the relation name.

relation-id

Indicates the relation ID as defined in the database.

Important! Set the relation-id properly. Otherwise, the correlation does not work.

Chapter 4: Creating and Managing Policies

This section contains the following topics:

[Create a Policy](#) (see page 29)

[Manage a Policy](#) (see page 29)

Create a Policy

A policy is a set of one or more rules. Create the policy or policies that contain the correlation rules to associate with the domain. Apply the policies to domains to create correlation domains.

Follow these steps:

1. [Open Condition Correlation Editor](#) (see page 13).

The Conditions tab is displayed by default.

2. Click the Policies tab.

The Condition Correlation Editor window displays a list of policies.

3. Click  (Create).

The Create Correlation Policy dialog opens.

4. Supply a value for each of the following policy properties:

Policy Name

Defines the policy (such as Power_Outage, DiskPolicy).

Policy Rule(s)

Includes the rules for the policy. You can use the arrow buttons to add rules from the Available Rules list to the Policy Rule(s) list, or to remove rules from the Policy Rule(s) list.

5. Click OK.

A new policy is created and added to the Policies tab list. The Author property identifies you as the author of the policy.


Manage a Policy

In the Condition Correlation Editor window, you can edit, copy, and delete a policy from the list of predefined and custom policies.

Follow these steps:


1. Click the Policies tab in the Condition Correlation Editor window.

The Condition Correlation Editor window displays a list of policies.

2. Select the policy to modify, and click  (Edit).

The Edit Policy dialog opens.

Note: If the policy is applied to a correlation domain, you cannot edit that policy name.


3. To copy a policy, click  (Copy).

The Copy Policy dialog opens, displaying the property settings for the policy you selected.

Note: The suffix _COPY is appended to the Policy Name to create a unique name for the new policy. A unique name is required.

4. Edit policy properties as necessary, and click OK.

The Condition Correlation Editor saves your changes.

5. To delete a policy, click  (Delete).

A confirmation dialog opens.

6. Click Yes.

The Condition Correlation Editor removes the selected policies from the Policies tab list.

Chapter 5: Creating and Managing Domains

This section contains the following topics:

[About Correlation Domains](#) (see page 31)

[Manage a Domain](#) (see page 33)

About Correlation Domains

You can create *correlation domains* that contain different correlation policies for various types of managed resources and alarm events. This section describes how to create correlation domains and edit domain settings.

Important! The volume of correlation processing that is required for large domains can affect CA Spectrum performance.

In the Condition Correlation Editor, you can create a domain, or you can copy a domain and modify it.

You can also create a domain from the context of a device, service, or Global Collection model that you want to add to the domain. Use the OneClick 'Add To' feature to create a domain in context.

Note: If you plan to add resources to the correlation domain from multiple landscapes, create the domain on the Main Location Server.

Create a Domain in the Condition Correlation Editor

In CA Spectrum, a *domain* is a group of resources. CA Spectrum Condition Correlation evaluates these resources collectively. You can apply policies to domains to create correlation domains. The rules in the policies that are applied to the domain are executed on all resources in the domain.


Follow these steps:

1. [Open Condition Correlation Editor](#) (see page 13).

The Conditions tab is displayed by default.

2. Click the Domains tab.

The Condition Correlation Editor window displays a list of any domains that users have created. Condition Correlation does not include default domains.

3. Click  (Create) to create a domain.

The Create Domain dialog opens.

4. Provide a value for each of the following domain properties:

Domain Name

Defines the domain (such as Backup_Power, DiskMonitor).

Landscape

Defines the landscape for the domain.

5. Move one or more policies from the Available Policies box to the Domain Policies box. If you are creating another version of an existing domain, remove policies as required from the Domain Policies box.
6. Add or remove resources from the domain by taking the following steps:

- a. Click the Resources tab, and click Create.

The Locate Resources dialog opens.

- b. Search for the resources to add to the domain in the 'Search using' panel.
- c. Select the resources to add to the domain from the search list, click 'Add Selected to Correlation Domain,' and click Close.

The resources that you added appear under the Resources tab in the Create Correlation Domain dialog.

Note: To add a device model and some port models for the device, add each individual model to the domain. Adding the device model does not add its component models to the domain.

7. Click OK.

The Condition Correlation Editor saves the new domain to the Domains tab list.

Create a Domain in the OneClick Console

You can use the OneClick 'Add To' feature to create a domain from the context of a device, service, or Global Collection model.

Follow these steps:

1. Select the model in OneClick that you want to use to create a domain.
2. Right-click the model, and select Add To Correlation Domain.
The Add to Correlation Domain dialog opens.
3. Perform one of the following actions:
 - To create a domain, enter a name for the domain and specify the landscape where you want to create the domain in the 'Create a new correlation domain' section.
 - To include the device, service, or Global Collection model in an existing domain, select the existing domain from the list in the 'Select an existing correlation domain' section.
4. Click OK.

A domain is created or edited. You can add policies to the domain. For more information, see [Create a Domain in the Condition Correlation Editor](#) (see page 32).


Manage a Domain

In the Condition Correlation Editor, you can copy and modify domains from the list of predefined and custom domains. You can delete domains that were created by any user.


Follow these steps:


1. Click the Domains tab in the Condition Correlation Editor window.

The Condition Correlation Editor displays a list of domains.

2. Select the domain to edit and click  (Edit).

The Edit Domain dialog opens.

3. To copy a domain, click  (Copy).
The Copy Domain dialog opens.
4. Edit the Domain properties as necessary, and click OK.
5. (Optional) Remove resources from a domain that you are copying by taking the following steps:
 - a. Select the resources that you want to remove from the Resources tab list in the Edit Correlation Domain dialog.
 - b. Click Delete.
The selected resources are removed from the Resources tab list.The Condition Correlation Editor saves your changes.

6. To delete a domain, click  (Delete).
7. Click Yes in the confirmation dialog that opens.
Condition Correlation Editor removes the selected domains from the Domains tab list.

Chapter 6: Testing and Debugging

CA Spectrum Condition Correlation provides advanced capabilities to enhance the functionality of the base product. When you create or customize the conditions, rules, and policies, that compose a condition correlation system, testing is required. We recommend staging the deployment of each new system to enable testing and debugging.

The topics in this section describe the testing and debugging process in Condition Correlation and recommend some best practices.

This section contains the following topics:

[How to Develop and Test Correlations](#) (see page 35)

[Testing a Correlation](#) (see page 36)

[Debugging Correlations](#) (see page 41)

How to Develop and Test Correlations

This section describes the design and development process for Condition Correlation. The lifecycle of a condition correlation follows typical software development methodology. Perform the following tasks in this recommended order to create a correlation:

- [Create conditions](#) (see page 17)
- [Create rules](#) (see page 23)
- [Create policies](#) (see page 29)
- [Create correlation domains and add models to the domains](#) (see page 32)

Note: Use the same order to develop new correlations using the Condition Correlation Editor.

After creating the condition correlation system, test and debug it. The testing and debugging process for a new correlation includes the following tasks:

1. Simulate the symptom condition on the appropriate model.
2. Verify that the appropriate alarm or event is raised in the OneClick Alarm View.

3. Verify that the condition is recognized by the correlation domain.
4. Simulate the root-cause condition of the appropriate model(s) in the correlation domain.
5. Verify that the condition is recognized by the correlation domain.
6. Verify that the root-cause alarm is raised correctly and that the symptom condition is hidden in the OneClick Alarm View.

Guidelines and Best Practices

Verify the following guidelines and best practices before starting the development process:

- Become familiar with the preconfigured Condition Correlation components, such as the conditions, rules, and policies that are installed with CA Spectrum. Use the correlation components when required. You can create copies and can edit the preconfigured correlation components. For more information, see [Condition Correlation Components](#) (see page 9).
- Start with simple conditions and rules to build a more complex system.
- Design the rules for easier testing. Use the techniques to test Condition Correlations that are explained in this document.
- Start testing rules and conditions from the bottom of the hierarchy and move up the hierarchy.

Testing a Correlation

After you design and develop a Condition Correlation system, test the system elements before deploying the correlation in a production environment. Multiple methods are available to perform the validation and verification process. The most robust method is to use a live environment. In certain circumstances, this method is not possible. For example, some CA Spectrum operators and developers cannot bring down infrastructure resources for testing a correlation.

A lab environment can provide a suitable test-bed for this type of verification. However, a lab can lack some of the resources that are required to test the correlation and simulate the scale of the deployment.

The simplest way to test a new Condition Correlation system is to create artificial events on CA Spectrum models. The following methods are available to test the Condition Correlation system:

- [Test the Correlation with the Command Line Interface](#) (see page 37)
- [Test the Correlation with the Web Services API](#) (see page 38)

Note: The Web Services API method provides more capability with greater complexity.

Test the Correlation with the Command-Line Interface

Use a simple command from the CA Spectrum Command-Line Interface (CLI) to test a new correlation. All events in CA Spectrum have an ID that is used to identify the event when it is processed. To know the type of the event that you are creating, an event-type-id is required. The event-type-id can be obtained from any of the following sources:

- The Condition Correlation Editor
 - Note:** The event is typically defined in an existing condition.
- The Event Configuration tool.
- The EventDisp file that refers the specific event and how it is handled.

Follow these steps:

1. Select Start, Programs, and Command Prompt.
The DOS prompt appears, ready to accept CLI commands.
2. Start the SpectroSERVER to which you want to connect.
3. Navigate to the following vnmsh directory in the CA Spectrum installation directory:

```
$ cd $SPECROOT/vnmsh
```
4. Open the connection using the following command:

```
$ connect
```

You are connected to the CLI session.

Note: On a UNIX platform, you can start a CLI session from the shell prompt. You can also start a CLI session from a bash shell prompt on Windows platform. For more information, see the *CA Spectrum Command Line Interface User Guide*.

5. Execute the following CLI command to test the correlation:

```
create event type=event-type-id text=event-text mh=model-handle
```

An event is created on the model with the given mode handle.

Example

To simulate the Chassis Down event on a model (with the model handle of 0x10234), use the following command:

```
create event type=0x10f69 text="Chassis is Down" mh=0x10234
```

Note: This command works for some situations. However, it does not let you deliver event variables in the event message. To generate a more complicated event, [use the web services method](#) (see page 38).

Test the Correlation with the Web Services API

Use the CA Spectrum RESTful Web Services API to test the correlation. The CA Spectrum RESTful Web Services lets you generate events that include event variables. This method requires a REST client that supports XML input, for example, [WizTools RESTclient](#) (see page 71) (on Windows 7).

Follow these steps:

1. Download and install the WizTools REST client.

For more information, see <http://code.google.com/p/rest-client/>.

2. Determine whether event variables are required to create the event.
3. Verify the syntax in the CsEvFormat file for the relevant event.

You can view this file through Event Configuration or by accessing the file directly, using bash shell or your preferred text editor.

For example, to find the file for event type 0x10f96, use the following path:

```
$SPECROOT/SG-Support/CsEvFormat/Event00010f96[language_pack]
```

Note: The language pack extension is used for CA Spectrum releases 9.3.0 or later. The extension for US English is '_en_US'.

4. Review the contents of this file. It does not contain event variables. It contains the following text:

The SpectroSERVER physical Memory has exceeded 2.5 Gigabytes for more than 300 seconds.

5. If you do not see event variables in the event message, use the template that lacks event variables. For more information, see [REST Examples for Correlation Testing](#) (see page 69).

6. For event type 0x5180302, verify the text for the following message:

The BGP Peering session from *S 1* to *S 2* has been Lost.

The event variables in italics are required for the event to be generated correctly. In this example, following event variables are applicable:

S 1

Represents the device model name.

S 2

Represents the Provider_Cloud model name.

The context of these parameters is determined by reviewing the actual events that have been generated.

7. Use the template to generate an event with the appropriate event variables. For more information, see [REST Examples for Correlation Testing](#) (see page 69).
An event is generated.

Verify the Simulated Events

You can verify the simulated events that are generated through the CLI or web services. After the test tools are configured, you can verify the events in the Event View for the target model.

Follow these steps:

1. Open the Condition Correlation Editor.

The Condition Correlation Editor window opens.

2. Select a CA Spectrum model in one of the views.

For example, select a view from the Navigation, List, Topology, or Search Results pane.

3. Select the Event tab for that model.

All events are displayed for the selected model.

4. Enter text in the event filter dialog to search for events that contain relevant text.

For example, if you are simulating a Border Gateway Protocol (BGP) backwards transition event, type 'BGP' as the keyword in the filter dialog.

5. Verify that the event is correctly displayed in the event window.

6. Verify that the event variables contain valid values.

Note: Perform Step 6 only if the correlation requires specific event variables to be set or modified. For more information, see [REST Examples](#) (see page 69).

The simulated event is verified.

Chapter 7: Debugging Correlations

Debugging is an essential component of the process to design, develop, and validate a new condition correlation system. Multiple built-in tools are available to help you debug a condition correlation. This section discusses the development prerequisites and the debugging tools that are available to debug the correlation system.

This section contains the following topics:

[Debugging Prerequisites](#) (see page 41)

[Debugging Tools](#) (see page 41)

Debugging Prerequisites

Review the following prerequisites before debugging a Condition Correlation system. In our testing, we have frequently seen preventable errors that are related to these factors:

- The correlation domain has at least one policy that is applied to it.
- The correlation domain contains the models where a correlation is likely to occur. Verify that the model has not been deleted from the correlation domain.
- Rules are set up with the correct relationships.
- Symptom conditions are set up with the correct event types.
- Symptom and root-cause conditions are not reversed.
- Rules or symptoms are set up correctly to suppress the alarms.
- Conditions or policies exist in the correlation domain.

Debugging Tools

Condition Correlation is one of the most complex systems in CA Spectrum. As a result, various tools and techniques are available to debug a correlation system. The Condition Correlation debugging tools are specific actions that are sent to specific models. The following model support actions are available for the Condition Correlation debugging tools:

- [Correlation Domain \(custom – possibly multiple per landscape\)](#) (see page 42)
- [Correlation Manager \(predefined – only one model per landscape\)](#) (see page 43)

You can send these actions through the CA Spectrum CLI or Web Services. Both APIs let you create actions on models with optional parameters. A prerequisite for each approach is to find the model handle of the target model. Use the following syntax for CLI-based actions:

```
update action=0xffff0102 mh=CorrelationDomain_mh
```

All debug output, whether it is initiated through the CA Spectrum CLI or Web Services, appears in the Control Panel message window. The debugging messages are captured in the following VNM.OUT file:

```
$SPECROOT/SS/VNM.OUT
```

Note: Before you attempt the debugging actions, we recommend verifying the events that are produced and checking the configuration of symptoms and rules. [Review the debugging prerequisites](#) (see page 41) before you start the process.

Debugging Actions: Correlation Domain

The following debug actions can be sent to the Correlation Domain Model through the CLI. You can see the output on the server console. The model handle of the target model is a prerequisite.

Action Code	Outputs	Usage
0xffff0102	List of existing conditions	Verify that expected conditions appear
0xffff0103	Detailed list of existing conditions	Verify that expected condition details match
0xffff0104	Condition definitions	Verify that expected conditions appear active
0xffff0105	Rule definitions	Verify that expected rules appear active
0xffff0106	Detailed Rule definitions	Verify that expected rule details match CCE
0xffff0107	Correlation hierarchy - models in domain	Verify that the target models are in domain
0xffff0202	Count conditions existing in domain	Verify that the count is an expected value
0xffff0203	Details for all count conditions	Verify that the count is an expected value
0xffff0900	Start runtime debug	Enable runtime debugging of domain
0xffff0901	Stop runtime debug	Disable runtime debugging

Note: Disable runtime debugging when it is not required to reduce the impact on performance and disk space.

Debugging Actions: Correlation Manager

The following debug actions can be sent to the Correlation Manager through the CLI. You can see the output on the server console. The model handle of the target model is a prerequisite.

Action Code	Outputs	Usage
0xffff0100	All condition definitions	Verify that expected condition appears
0xffff0101	All condition definitions – detailed	Verify that expected condition details match
0xffff0110	All rule patterns	Verify that expected rule appears
0xffff0111	All rule patterns – detailed	Verify that expected rule details match
0xffff0120	All policies	Verify that expected policy appears
0xffff0200	All condition registrations	Verify that target model registrations are present
0xffff0300	Condition engine condition table	Verify that active conditions are represented
0xffff0401	Reload shipped condition definitions	Restore initial conditions
0xffff0402	Reload shipped rule definitions	Restore initial rules
0xffff0403	Reload shipped policy definitions	Restore initial policies
0xffff0900	Enable Condition Correlation Mgr debug	Verify initialization, registration, and notification
0xffff0901	Disable Condition Correlation Mgr debug	Disable when not required
0xffff0910	Enable condition engine debug	Verify event and alarm registrations received
0xffff0911	Disable condition engine debug	Disable when not required

Note: Disable runtime debug when it is not required to reduce the impact on performance and disk space.

Appendix A: Condition Correlation Examples

This appendix provides a workflow and examples to help you implement Condition Correlation in your environment.

Note: All of the fictitious instances of alarms and Condition Correlation components that are referenced in the following examples are enclosed in double quotation marks (" "). References to actual events and alarms that are defined in CA Spectrum are not enclosed in quotation marks.

The following scenarios are discussed as Condition Correlation examples:

- Power Outage
- Disk Full
- WAN Link Failure

How to Configure a Condition Correlation for a Power Outage

CA Spectrum Condition Correlation can be configured to determine the root-cause alarm and manage trouble-prone segments of your infrastructure. Predefined correlation systems are available in Condition Correlation Editor. However, you can use Condition Correlation to select the criteria that identify a causal problem event. With the help of Condition Correlation components (such as Conditions, Rules, Policies, and Correlation Domains), you can pinpoint the root-cause alarm and can pay less attention to symptomatic alarms.

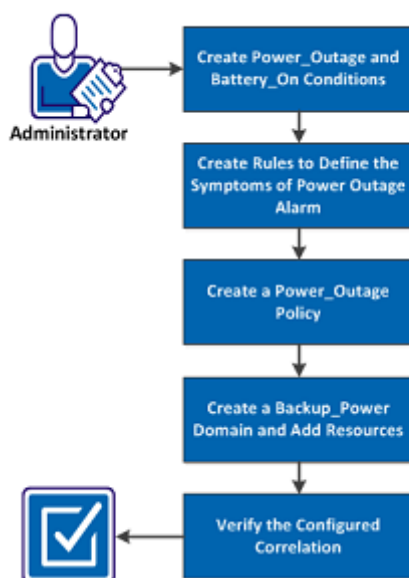
CA Spectrum includes many predefined correlations. For example, ContactLost_Red (caused by) Chassis Down, LinkDown (caused by) Chassis Down, Dev Module Pulled (caused by) Blade Status Unknown are a few predefined correlations that are available in Condition Correlation Editor.

In a power outage scenario, managed UPS systems generate traps indicating that they have switched to backup battery power. If the backup battery power fades, the systems generate traps that indicate low battery power. When the batteries fail, managed devices that are connected to the UPS systems go down. These devices trigger a flood of events and alarms from the affected area. The volume of events makes it difficult to identify and address the underlying problem.

As an administrator, you can configure a correlation system for a power outage. Create one or more conditions that can be evaluated by correlation rule criteria. If rule criteria are met, Condition Correlation identifies one condition as the root-cause condition and the other condition as symptomatic of the root-cause condition. Create a policy that contains the correlation rules to associate with the domain and apply the policy to the domain. The Condition Correlation process is in effect for the resources that are included in the domain

The following diagram illustrates the process to configure a Condition Correlation for a power outage:

How to Configure a Condition Correlation for a Power Outage



Perform the following tasks to configure a condition correlation for a power outage:

1. [Create Power_Outage and Battery_On Conditions](#) (see page 47)
2. [Create Rules to Define the Symptoms of Power Outage Alarm](#) (see page 48)
3. [Create a Power_Outage Policy](#) (see page 50)
4. [Create a Backup_Power Domain and Add Resources](#) (see page 50)
5. [Verify the Configured Correlation](#) (see page 51)

Create Power_Outage and Battery_On Conditions

Conditions are the building blocks of the correlation system. Create Power_Outage and Battery_On conditions in the Condition Correlation editor to configure a correlation system for a power outage. The Power_Outage condition uses the set event code and the clear event code that are associated with the Power Outage alarm. The Battery_On condition uses the set event code and the clear event code that are associated with the UPS trap. You can also use this procedure to create conditions to handle alarms from other root causes.

Note: To access Condition Correlation Editor, you require OneClick administrative privileges.

Follow these steps:

1. Open Condition Correlation Editor.

The Condition Correlation Editor window opens. For more information, see the *Condition Correlation User Guide*.

2. Click the Conditions tab.

A list of conditions is displayed.

3. Click  (Create).

The Create Correlation Condition dialog opens.

4. Specify a value for the following condition properties:

Condition Name

Defines the condition. For example, supply the names Power_Outage and Battery_On.

Set Event Code

Identifies the CA Spectrum event code that is associated with the condition. For example, use the following set event codes:

- Set event for Battery_On: 0x0116905a
- Set event for Power_Outage: 0x01169431

5. Click OK.

Power_Outage and Battery_On conditions are created.

Create Rules to Define the Symptoms of the Power Outage Alarm

A rule is defined to stipulate that one condition is a symptom or a cause of another condition. Create rules to define the symptoms of the Power Outage alarm. You can create the following three rules for Power Outage:

Battery On -> Power Outage

Specifies that if five or more power systems go on battery power, the Power Outage condition is the implied cause.

ContactLost_Red -> Power Outage

Specifies that if the ContactLost_Red condition (predefined) is caused by the Power Outage condition, the critical (red) Contact Lost alarm is suppressed as a symptom of the Power Outage alarm.

ContactLost_Gray -> Power Outage

Specifies that if the ContactLost_Gray condition (predefined) is caused by the Power Outage condition, the (gray) Contact Lost alarm is suppressed as a symptom of the Power Outage alarm.

Follow these steps:

1. Open Condition Correlation Editor.

The Conditions tab is displayed by default.

2. Click the Rules tab.

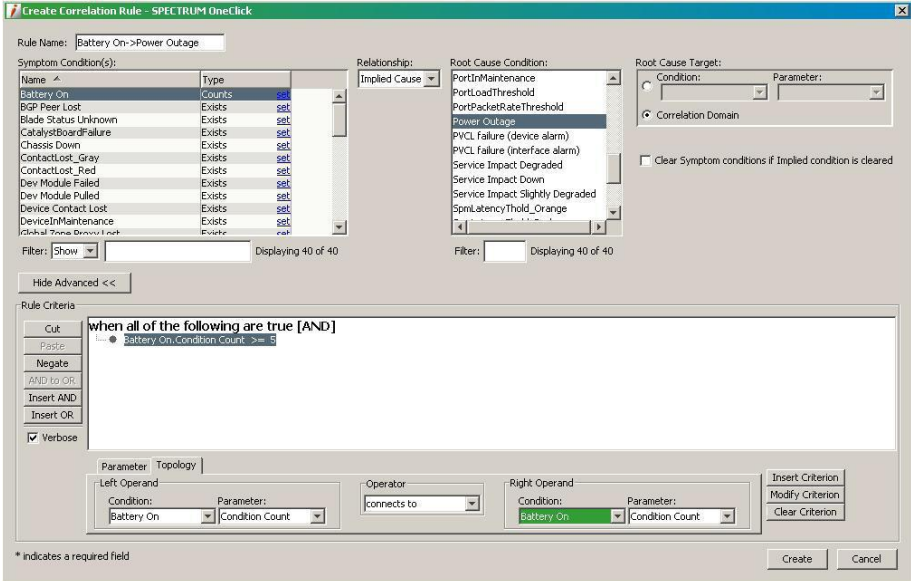
A list of rules is displayed.

3. Click  (Create).

The Create Rule dialog opens.

4. Enter a name for the rule in the Rule Name field.

The following image illustrates the configuration in Advanced Rule Criteria:



Create Correlation Rule - SPECTRUM OneClick

Rule Name:

Symptom Condition(s):

Name	Type	Count
Battery On	Counts	set
BGP Peer Lost	Exists	set
Blade Status Unknown	Exists	set
CatalystBoardFailure	Exists	set
Chassis Down	Exists	set
ContactLost_Gray	Exists	set
ContactLost_Red	Exists	set
Dev Module Failed	Exists	set
Dev Module Pulled	Exists	set
Device Contact Lost	Exists	set
DeviceInMaintenance	Exists	set
External Probe Device Fault	Exists	set

Relationship:

Root Cause Condition:

PortInMaintenance
PortLoadThreshold
PortPacketRateThreshold
Power Outage
PVCL Failure (device alarm)
PVCL Failure (interface alarm)
Service Impact Degraded
Service Impact Down
Service Impact Slightly Degraded
SpmlatencyThold_Orange

Root Cause Target:

Clear Symptom conditions if Implied condition is cleared ☐

Filter: Displaying 40 of 40

Hide Advanced <<

Rule Criteria

Cut Paste Negate AND to OR Insert AND Insert OR Verbose

when all of the following are true [AND]

- Battery On Condition Count >= 3

Parameter Topology

Left Operand: Condition: Parameter:

Operator:

Right Operand: Condition: Parameter:

Insert Criterion Modify Criterion Clear Criterion

* indicates a required field

Create Cancel

5. Click Create.

Rules for Power Outage are created in Correlation Editor.

Create a Power_Outage Policy

A policy is a set of one or more rules. To configure a condition correlation system for a Power Outage, create a Power_Outage policy and add the Power Outage rules to the Policy rules list. All implementations of the policy are updated after you add rules.

Follow these steps:

1. Open Condition Correlation Editor.
The Conditions tab is displayed by default.
2. Click the Policies tab.
The Condition Correlation Editor window displays a list of policies.



3. Click (Create).
The Create Correlation Policy dialog opens.
4. Supply a value for each of the following policy properties:

Policy Name

Defines the policy. For example, supply the name Power_Outage.

Policy Rule(s)

Includes the rules for the policy. You can use the arrow buttons to add rules from the Available Rules list to the Policy Rule(s) list, or to remove rules from the Policy Rule(s) list.


A Power_Outage policy is created that includes the Battery On -> Power Outage, ContactLost_Red -> Power Outage, and ContactLost_Gray -> Power Outage rules.

Create a Backup_Power Domain and Add Resources

A *domain* is a group of resources. You can create a Backup_Power domain for the condition correlation system for a Power Outage. UPS models and the device models (that connect to the power supplies) are the resources of the correlation domain. These resources are added and the Power_Outage policy is applied to the domain. You can include multiple models of various model types and can apply multiple policies.

Follow these steps:

1. Open Condition Correlation Editor.
The Conditions tab is displayed by default.
2. Click the Domains tab.
The Condition Correlation Editor window displays a list of any domains that users have created. Condition Correlation does not include default domains.

3. Click  (Create) to create a domain.

The Create Domain dialog opens.

4. Provide a value for each of the following domain properties:

Domain Name

Identifies the domain. For example, supply the name Backup_Power.

Landscape

Defines the landscape for the domain.

5. To add the Power_Outage policy to the Domain Policies box, move the Power_Outage policy from Available Policies to the Domain Policies box.

If you are creating another version of an existing domain, remove policies as required from the Domain Policies box.

The following image illustrates the Power_Outage policy that is added to the Domain Policies box.



6. Click the Resources tab to add or remove resources from the domain. For more information, see the *CA Spectrum Condition Correlation User Guide*.
7. Click OK.

The Backup_Power domain is created.

Verify the Correlation

As a best practice, verify the correlation that you configured before you deploy it. When a correlation is correctly configured, the symptom alarms are hidden while the root cause alarm is active.

Follow these steps:

1. Log in to the OneClick console.
2. Click the Alarms tab.

The Alarms window opens.

3. Verify the status of the symptom alarms (such as ContactLost_Red, ContactLost_Red, Battery_On) and the root cause alarm.

The symptom alarms are hidden and the root cause alarm is displayed in Alarms tab.

The correlation is appropriately configured.

Disk Full Scenario

A disk monitor alarm appears on many models multiple times. However, the total number of these alarms is required rather than every instance of each alarm. For an instance, less than five disk monitor alarms are acceptable, but once there are at least five alarms, you want to see a minor alarm. Similarly, if there are more than ten alarms, you want to see a major alarm. If there are more than 15 alarms, then you can see a critical alarm. The following process explains the concept of condition correlation for a Disk Full scenario:

- If any of the DiskFull events are generated on the host devices with different values for the disk (variable binding 4), they are displayed on-screen, as long as their overall number goes up to four. Once a fifth alarm is generated on a model of the correlation domain, the MinorDiskProblemRule instantiates and the MinorDiskProblem alarm is created on the correlation domain. The five DiskFull alarms are hidden as symptoms under the MinorDiskProblem alarm.
- If one or more DiskFull alarms are cleared, the MinorDiskProblem alarm clears, showing the previously hidden other four or fewer DiskFull alarms. By contrast, if more DiskFull alarms are generated and their number reaches ten, the MajorDiskProblem alarm is generated. The minor alarm, which covers 5-9 alarms, disappears. All DiskFull alarms can be the symptoms of the major alarm.
- If the DiskFull alarm total does not reach ten, you can see the MinorDiskProblem alarm. Similarly, if the DiskFull alarm total exceeds 14, you can see the CriticalDiskProblem alarm.

EventDisp Entries

You can use the following EventDisp entries for setting up Condition Correlation. These alarms use variable binding 4 as a discriminator so that multiple alarms can exist on the same device.

test alarm (disk full)

0xffff0000 E 50 A 1,0xffff0000,4

0xffff0001 E 50 C 0xffff0000,4

5 to 9 test alarms, minor problem with disks

0xffff0010 E 50 A 1,0xffff0010

0xffff0011 E 50 C 0xffff0010

10 to 14 test alarms, major problem with disks

0xffff0020 E 50 A 2,0xffff0020

0xffff0021 E 50 C 0xffff0020

more than 15 test alarms, critical problem with disks

0xffff0030 E 50 A 3,0xffff0030

0xffff0031 E 50 C 0xffff0030

Note: You can create event format and alarm probable cause files. For more information, see the *Event Configuration User Guide*.

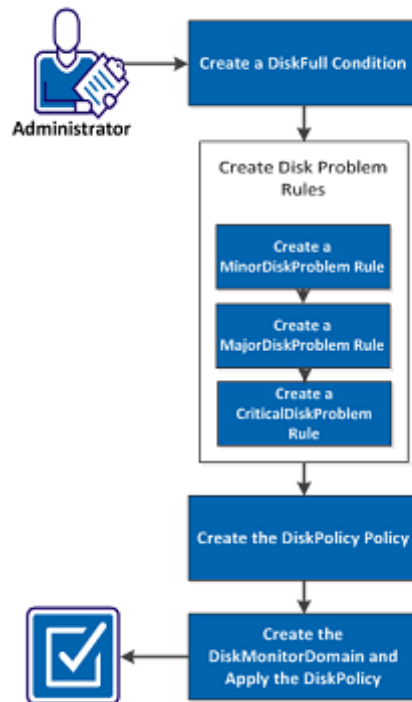
How to Configure the Sample DiskFull Condition Correlation

Condition Correlation determines the root-cause alarm by selecting a criterion to identify a casual problem event. You can use the predefined components to configure a correlation. With the help of Condition Correlation components (such as Conditions, Rules, Policies and Correlation domain), you can pinpoint the root-cause alarm and symptomatic alarms.

As an administrator, you can configure the sample Disk Full Condition Correlation by creating DiskFull condition and Disk Problem rules. After creating the conditions and rules, you can create a DiskPolicy policy and apply it to a DiskMonitor Domain. DiskFull events are generated on the host devices with different values for the disk. Each time the disk problem alarms are cleared, all the existing DiskFull alarms become symptoms of the respective disk problem alarm.

The following diagram illustrates the process to configure a Disk Full Condition Correlation:

How to Configure a Sample DiskFull Condition Correlation



Perform the following tasks to configure the sample DiskFull Condition Correlation:

1. [Create a DiskFull Condition](#) (see page 55)
2. [Create Disk Problem Rules](#) (see page 56)
 - a. [Create a MinorDiskProblem Rule](#) (see page 56)
 - b. [Create a MajorDiskProblem Rule](#) (see page 58)

- c. [Create a CriticalDiskProblem Rule](#) (see page 59)
3. [Create the DiskPolicy Policy](#) (see page 60)
4. [Create the DiskMonitorDomain and Apply the DiskPolicy](#) (see page 60)

Create Disk Conditions

You can create disk conditions in Condition Correlation Editor. You can specify the condition name with the set event and clear event codes associated to the condition.

Follow these steps:

1. [Open Condition Correlation Editor](#) (see page 13).

The Condition Correlation Editor window opens.

2. Click the Conditions tab.

A list of predefined and user-created conditions is displayed.

3. Click Create.

The Create Correlation Condition dialog opens.

4. Create the following disk conditions:

- DiskFull Alarm (including Disk parameter):

- Condition Name: DiskFull
- Set Event Code: 0xffff0000
- Clear Event Code: 0xffff0001

Because a model can have multiple occurrences of these conditions, you must also add a parameter to distinguish them. The alarms are distinguished by variable binding 4. Therefore, use 4 for this parameter as well.

- Parameter Name: Disk
- Parameter Type: Var Bind
- Parameter ID: 4
- Use as discriminator: Yes

- Minor Disk Problem:

- Condition Name: MinorDiskProblem
- Set Event Code: 0xffff0010
- Clear Event Code: 0xffff0011

- Major Disk Problem:
 - Condition Name: MajorDiskProblem
 - Set Event Code: 0xffff0020
 - Clear Event Code: 0xffff0021
 - Critical Disk Problem:
 - Condition Name: CriticalDiskProblem
 - Set Event Code: 0xffff0030
 - Clear Event Code: 0xffff0031
5. Click OK.
- Conditions are created and added to the Conditions tab.

Create Disk Problem Rules

You can create Disk Problem Rules to configure a Disk Full Condition Correlation. Minor, Major, and Critical Disk Problem rules are created in Condition Correlation Editor with specific rule criteria.

You can create the following disk problem rules:

- [Create a MinorDiskProblem Rule](#) (see page 56)
- [Create a MajorDiskProblem Rule](#) (see page 58)
- [Create a CriticalDiskProblem Rule](#) (see page 59)

Create a MinorDiskProblem Rule

You can create a MinorDiskProblem rule in Condition Correlation Editor. You can specify the rule name and the rule criteria for the DiskFull condition.

Follow these steps:

1. [Open Condition Correlation Editor](#) (see page 13).
The Condition Correlation Editor window opens.
2. Click the Rules tab.
A list of rules is displayed
3. Create a rule using the following properties:
 - Name: MinorDiskProblemRule
 - Symptom Condition(s):
 - Name: DiskFull
 - Type: Counts

- Relationship: Implied Cause

The MinorDiskProblem alarm is generated when the rule criteria are satisfied, and causes the rule to hide the DiskFull alarms.
 - Root Cause Condition: MinorDiskProblem
 - Root Cause Target: Select the Correlation Domain option.
4. Click Show Advanced to open the Rule Criteria panel.
 5. Create the following rule criteria:
 - 'DiskFull.count GREATER THAN OR EQUAL TO 5':
 - Condition: DiskFull
 - Parameter: Condition Count
 - Operator: GREATER THAN OR EQUAL TO
 - By Value: Yes
 - Value: 5
 - Type: Integer
 - Click Insert Criterion.
 - 'DiskFull.count LESS THAN 10':
 - Condition: DiskFull
 - Parameter: Condition Count
 - Operator: LESS THAN
 - By Value: Yes
 - Value: 10
 - Type: Integer
 6. Click Insert Criterion.
 7. Click Create.

The new rule is added to the Condition Correlation Editor Rules tab.

Create a MajorDiskProblem Rule

You can create a MajorDiskProblem rule for the DiskFull condition in Condition Correlation editor. After creating the rule, you can specify the rule criteria.

Follow these steps:

1. [Open Condition Correlation Editor](#) (see page 13).

The Condition Correlation Editor window opens.

2. Click the Rules tab.

A list of rules is displayed.

3. Create a rule using the following properties:

- Name: MajorDiskProblemRule
- Symptom Condition(s):
 - Name: DiskFull
 - Type: Counts
- Relationship: Implied Cause
- Root Cause Condition: MajorDiskProblem
- Root Cause Target: Select the Correlation Domain option.

4. Click Show Advanced to open the Rule Criteria panel.

5. Create the following rule criteria:

- 'DiskFull.count GREATER THAN OR EQUAL TO 10':
 - Condition: DiskFull
 - Parameter: Condition Count
 - Operator: GREATER THAN OR EQUAL TO
 - By Value: Yes
 - Value: 10
 - Type: Integer
 - Click Insert Criterion.
- 'DiskFull.count LESS THAN 15':
 - Condition: DiskFull
 - Parameter: Condition Count
 - Operator: GREATER THAN OR EQUAL TO
 - By Value: Yes

- Value: 15
 - Type: Integer
6. Click Insert Criterion.
 7. Click Create.

The new rule is added to the Condition Correlation Editor Rules tab.

Create a CriticalDiskProblem Rule

You can create the CriticalDiskProblem rule after creating MinorDiskProblem and MajorDiskProblem rules in Condition Correlation Editor.

Follow these steps:

1. [Open Condition Correlation Editor.](#) (see page 13)
The Condition Correlation Editor window opens.
2. Click the Rules tab.
A list of rules is displayed
3. Create a rule using the following properties:
 - Name: CriticalDiskProblemRule
 - Symptom Condition(s):
 - Name: DiskFull
 - Type: Counts
 - Relationship: Implied Cause
 - Root Cause Condition: CriticalDiskProblem
 - Root Cause Target: Select the Correlation Domain option.
4. Click Show Advanced to open the Rule Criteria panel.
5. Create the following rule criteria:
 - 'DiskFull.count GREATER THAN OR EQUAL TO 15':
 - Condition: DiskFull
 - Parameter: Condition Count
 - Operator: GREATER THAN OR EQUAL TO
 - By Value: Yes
 - Value: 15

- Type: Integer

6. Click Insert Criterion.
7. Click Create.

The new rule is added to the list of rules in Condition Correlation Editor.

Create the DiskPolicy Policy

You can create a DiskPolicy policy in Condition Correlation Editor. After creating the policy, you can add the disk problem rules to the policy rules list.

Follow these steps:

1. [Open Condition Correlation Editor.](#) (see page 13)

The Condition Correlation Editor window opens.

2. Click the Policies tab.

The Condition Correlation Editor window displays a list of policies

3. Create a new policy using the name DiskPolicy.
4. Add the following rules to the Policy Rules list:

- MinorDiskProblemRule
- MajorDiskProblemRule
- CriticalDiskProblemRule

5. Click Create.

The new policy appears in the list of policies in Condition Correlation Editor.

Create the DiskMonitorDomain and Apply the DiskPolicy

You can create a new correlation domain to accommodate the components. You can apply the DiskPolicy from the policies list to the DiskMonitor Domain.

Follow these steps:

1. Click the Domain tab in Condition Correlation Editor.
2. Click Create.
3. Enter DiskMonitorDomain in the Domain Name text box.
4. Click the Policies tab, select DiskPolicy from the Available Policies list and move it into the Domain Policies list.
5. Click the Resources tab and then select any number of host devices as resources.
6. Click Create.

The DiskMonitor Domain is added to the list of Domains in the Condition Correlation Editor.

Create a Clear Events Correlation

This section describes some additional functionality that you can implement on this sample DiskFull Condition Correlation.

You can clear the disk problem alarms (such as, major, minor, or critical) from OneClick. However, if you clear the disk problem alarm, all previously hidden disk full alarms reappear. Because the alarm it is correlated to is destroyed. This section describes how you can clear all of these alarms.

Multiple alarms on multiple models can exist. Therefore, the only thing you have is the clear event for one of the disk problem events (minor, major, or critical). You can perform the following tasks to create the clearing events on the correct model:

- [Create an Additional Parameter for the DiskFull Condition](#) (see page 61)
- [Create an Event Rule to Identify a Cleared Disk Problem Alarm](#) (see page 62)
- (Optional) [Log and Add an Event to Clear the DiskFull Alarms](#) (see page 62)
- Add an Event to Clear the Disk Full Alarms
- [Create the Conditions Required for the Clear Correlation](#) (see page 63)
- [Create a Rule to Clear DiskFull Alarms](#) (see page 64)

Create an Additional Parameter for the DiskFull Condition

To add a parameter to the DiskFull condition, you require one additional parameter for the DiskFull condition.

Follow these steps:

1. In the Conditions tab, in the Correlation Editor, select the DiskFull condition and click Edit.
The Edit Correlation Condition dialog opens.
2. Click Create in the Parameters section.
The Create Correlation Parameter dialog opens. You need to add the model where the condition (alarm) exists as shown in the following step.
3. From the Parameter Type field, select Predefined.
The Parameter ID field shows the applicable model handle attribute: 0x129fa.
4. Click Create to add this parameter to the condition.
This condition parameter can now be used in the clearing rule that you create to assert the clear event on the correct model.

Create an Event Rule to Identify a Cleared Disk Problem Alarm

You can create an event rule to identify when a disk problem alarm is cleared by the user. This rule lets you distinguish between instances when the correlation cleared the alarm (performs automatically when the number of alarms reaches any of the thresholds) and when a user cleared the alarm from the UI, indicating that the user knows about the problem and decides that the problem is resolved.

As you are clearing the alarms from the UI, no direct event code (for example, 0xffff0021) is used. Instead, you can use one of the alarm status events. For example, 0x10706: user has cleared an alarm. In this event, you can find Probable Cause Code of the cleared alarm, in varbind 3. You can use the Probable Cause Code to generate a new event and can use it as a condition to start the clear correlation.

You can create an event rule to generate a event, disk problem alarm has been user-cleared. The 0x10706 event is mapped (by default) in the following file:

```
<$SPECROOT>/SS/CsVendor/Cabletron/EventDisp
```

The syntax to add an event action is as follows:

```
0x00010706 E 50 R CA.EventCondition, \
    " { v 3 } = { H 0xffff0010 } ", 0xffff0100, \
    " { v 3 } = { H 0xffff0020 } ", 0xffff0100, \
    " { v 3 } = { H 0xffff0030 } ", 0xffff0100
```

Log and Add an Event to Clear the DiskFull Alarms

Optionally, you can log the event in the custom EventDisp file using the following syntax:

```
0xffff0100 E 50
```

You can add an event to clear the disk full alarms, regardless of their discriminator value. Use the following clear all ('A') alarm clear flag syntax:

```
0xffff0002 E 50 C 0xffff0000, A
```

This event lets you clear all disk full alarms on a model, even if you do not know the values for their discriminator attributes.

Create the Conditions Required for the Clear Correlation

Reload the EventDisp files so that you can set up the clear correlation. The following procedure describes the steps.

Follow these steps:

1. [Open Condition Correlation Editor](#) (see page 13).

The Condition Correlation Editor window opens.

2. Click the Conditions tab.

A list of conditions is displayed.

3. Click Create.

The Create Correlation Condition dialog opens.

4. Create the following condition to start the clear correlation:

- **DiskProblemAlarmUserCleared:**
 - Condition Name: DiskProblemAlarmUserCleared
 - Set Event Code: 0xffff0100
 - Clear Event Code: 0xffff0100

Note: The DiskProblemAlarmUserCleared condition is no longer required after it starts the clear correlation. You can clear this condition after the clear correlation has completed. You can use the same clear event as the set event to generate the condition. This condition is cleared after the completion of clear correlation and is therefore temporary.

5. Create the following condition to clear the DiskFull alarms:

- **DiskFullAlarmClear:**
 - Condition Name: DiskProblemAlarmUserCleared
 - Set Event Code: 0xffff0002
 - Clear Event Code: 0xffff0002

(The Set Event Code indicates that this event can be generated when the condition is generated by an implied rule.)

(The Clear Event Code indicates that the condition is self-clearing, as does the condition in Step 1.)

DiskProblemAlarmUserCleared condition is created.

Create a Rule to Clear DiskFull Alarms

You can create a rule that clears all DiskFull alarms when a user clears one of the disk problem alarms.

Follow these steps:

1. [Open Condition Correlation Editor](#) (see page 13).

The Condition Correlation Editor window opens.

2. Click the Rules tab.

A list of rules is displayed.

3. Create the following rule:

- Name: DiskFullUserClearRule
- Symptom Condition(s):
 - DiskProblemAlarmUserCleared
 - DiskFull
- Type: Exists
- Relationship: Implies
- Root Cause Condition: DiskFullAlarmClear
- Root Cause Target: DiskFull.Model

Note: This rule ensures that the clear event is generated on each model where a DiskFull alarm exists, enabling the alarm to be cleared.

4. Save the rule.
5. Add the new rule to the “DiskPolicy” policy.

This rule triggers when any one of the three disk problem alarms is cleared by the user, generating event 0xffff0100.

The setup is complete. Anytime the user clears any one of the three disk problem alarms (minor, major, or critical), all individual DiskFull alarms are cleared. The condition is paired with each DiskFull alarm, and generates the DiskFullAlarmClear condition on the model of the DiskFull alarm. Thus, all of the DiskFull alarms are cleared.

WAN Link Failure Example

The WAN Link Failure example describes the process that Condition Correlation uses to pinpoint the root-cause alarm and symptomatic alarms among a barrage of alarms that are generated for different resources, as a result of a WAN link failure.

WAN Link Scenario

In many WANs, primary connections have a backup. The backup connection typically provides less bandwidth than the primary connection. In this example, a 384K Frame Relay link is backed up by a 128K ISDN link. Also, a CA Spectrum Service Performance Manager (SPM) test is measuring latency across the WAN link.

When the Frame Relay link goes down, the ISDN link takes over and the SPM test exceeds the latency threshold due to the reduced bandwidth. CA Spectrum generates two alarms (Critical Alarm - Frame Relay Link Down occurs on the Frame Relay link model; Minor Alarm - SPM Test Exceeded Threshold occurs on the SPM Test model) and one event (ISDN Backup Active occurs on the device).

WAN Link Correlation Strategy

It may not be apparent to network management personnel that all the three conditions are related and they are likely to focus their effort on the critical alarm even though there are other important alarms in the infrastructure.

You can apply the following conditions to the domain including the resources that can be compromised by a primary WAN link failure:

- The two alarms and the ISDN event can be correlated to generate a new primary link down, reduced bandwidth condition that produces a major alarm because the WAN is still working, but with a decreased performance.
- The failed Frame Relay link can be correlated with the Dialup Link Active event and can imply that the primary WAN link is down with reduced bandwidth, if the backup link bandwidth is less than the primary link bandwidth.
- The SPM test can be correlated with the primary WAN link down, reduced bandwidth condition with a rule that stipulates SPM Test Threshold Exceeded is caused by primary WAN link down, reduced bandwidth condition.

This correlation system produces the following alarm and event information: the single major alarm for the WAN link being down. There is also an SPM test threshold exceeded alarm and the ISDN backup active event, but these alarms are hidden under the single major alarm. This lets troubleshooters focus their efforts on the most important alarms. A second rule could be created to produce a minor alarm if the active backup link has the same bandwidth as the failed primary link.

WAN Link Failure Configurations

You can configure the correlation system using the following process:

- In CA Spectrum, a new “Primary WAN Link Down, Reduced Bandwidth” alarm is created. A set event and clear event for the new alarm is required.

You can create alarms and can edit event configuration files. For more information, see the *Event Configuration User Guide*.

- Create the following conditions:
 - A “Primary_WAN_Link_Down_Reduced_Bandwidth” condition using the set and clear event codes from the “Primary WAN Link Down, Reduced Bandwidth” alarm.
 - A “Dialup_Link_Active” condition using set event 0x022ffff6, “Dialup link has been activated,” and clear event 0x022ffffc, “Dialup link is inactive.” This condition is not linked to a CA Spectrum alarm. However, it infers that the backup, or secondary, link is up and running.
- Create the following rules:
 - A “PrimaryFrameRelay_Red -> LinkDown” rule states that if the “Primary_WAN_Link_Down_Reduced_Bandwidth” and “Dialup_Link_Active” conditions occur, then the implied cause is the “Primary_WAN_Link_Down_Reduced_Bandwidth” condition and the critical (red) Frame Relay Link Down alarm is suppressed by the “Primary WAN Down, Decreased Bandwidth” alarm (orange).
 - An “SPMLatencyThreshold_yellow -> Violated” rule states that the SPM latency threshold violation is caused by the “Primary WAN Link Down, Reduced Bandwidth” condition, and suppressed yellow SPM latency threshold violation alarms.
- A “WAN_Link_Failure” policy is created that includes the “PrimaryFrameRelay_Red -> LinkDown” and “SPMLatencyThreshold_yellow -> Violated” rules.
- A “WAN_Primary_Backup_Links” domain is created. It includes the primary WAN link interfaces, the backup link, and any SPM tests that can be impacted by the lower bandwidth of the backup. The “WAN_Link_Failure” policy is applied to the domain.

Appendix B: Special Topics

This section discusses special topics that are related to Condition Correlation capabilities and implementation.

Condition Correlation and Fault Isolation

If a managed device stops responding to polls, the CA Spectrum fault isolation algorithm determines whether to create a critical alarm for the device or suppress its alarm state. The unreachable device is the root cause of the alarm. Condition Correlation supports setting up a correlation between a device in the Contact Lost condition and some other condition in your environment. For example, CA Spectrum receives a trap from a BGP router that is reporting a lost session with a peer router. If the peer router is already in the Contact Lost state in CA Spectrum, the BGPLost Session alarm can be a symptom of the Contact Lost alarm on the peer router model.

If the peer router in the Contact Lost state has a critical Device Has Stopped Responding to Polls alarm, the correlation is trivial. If the fault state of the peer router is suppressed by the CA Spectrum fault isolation algorithm, no root cause alarm exists on this model.

You cannot correlate the actual root cause alarm with the Peer Lost alarm without special consideration from Condition Correlation. However, in Condition Correlation, the Device Contact Lost condition receives special consideration. This condition remains in force whenever a device is in the Contact Lost state, regardless of whether the device model is suppressed or has an alarm. If the device model in question is suppressed, the correlation engine locates the isolated alarm and uses it as the root cause for any correlation rules.

About Transfer Rules

CA Spectrum recognizes a Model Active condition that can be used in a correlation rule. The Model Active condition is used when a port model is added or removed from a correlation domain. This condition can be used for special rules, such as transferring alarms from devices to ports, because the Model Active condition is present for each correlated port. This usage eliminates the requirement for the port to have an alarm to participate in a correlation. Attributes on the condition can then be used to create a rule that transfers alarms to the correct port. The correct port can be identified by using the following parameters from the Model Active condition:

- Component OID of the port.
- Model handle of the parent device of the port.
- Model type of the port.

Condition Correlation provides a default transfer rule: transfer PVCL alarm from the device to interface. It reacts to the PVCL failure condition (alarm 0x210048 - PVCLs Failure Notification) on the device model by extracting the Interface ID of the affected port from the PVCL failure condition. Then it locates the port model by comparing the Interface ID from the PVCL failure condition with the Component_OID parameter of the Model Active condition on the port. A new PVCL failure (0x210c0c - PVCLs Failure Notification) alarm is created on the port. This failure alarm is identified by the Model parameter of the Model Active condition. The PVCLs Failure Notification alarm for the device is made a symptom of the new PVCLs Failure Notification alarm on the port.

Advanced Correlations and Data Type Comparisons

Verify the following information before you configure advanced correlations, which involve comparisons between different data types:

- Condition Correlation converts the right-hand value to the left. This conversion can be problematic; it is unlikely that a real number conversion produces the same text string that you have for a comparison.
- SNMP represents both real text strings (such as, messages, and information), and octet strings (such as, Mac addresses) with no indication of the actual usage. Therefore, in some cases it is impossible for the automatic conversion process to convert to the actual type which you need for a comparison. Because, Condition Correlation does not have the meta-information.
- Condition Correlation does not attempt to convert list types.

Appendix C: REST Examples for Correlation Testing

This appendix contains resources to help you test and debug custom correlation systems that you have created.

Two examples of simulated events are provided to help you test correlations. In the following XML examples, replace the [model_handle] field with the actual model handle of the target model. You can perform this task through the CA Spectrum CLI (using show models) or through the attribute browser, by reading the value of the Model_Handle attribute with attribute ID of 0x129fa.

RESTful Web Services XML Example – No Event Variables

The following XML example lets you generate an event for testing purposes without event variables (for example, Event00010220, the model has gone into Maintenance Mode). Use this event template as a framework to develop simulation and testing tools for Condition Correlation testing.

```
<?xml version="1.0" encoding="UTF-8"?>
<rs:event-request throttlesize="10"
xmlns:rs="http://www.ca.com/spectrum/restful/schema/request"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ca.com/spectrum/restful/schema/request
../../../../xsd/Request.xsd">
<rs:event>
<!-- target model of event -->
<rs:target-models>
<rs:model mh="0x100000"/>
</rs:target-models>
<!-- event ID -->
<rs:event-type id="0x10220"/>
</rs:event>
</rs:event-request>
```

RESTful Web Services XML Example – with Event Variables

The following sample XML generates an artificial event for testing with event variables. Use this event template as a framework to develop simulation and testing tools for Condition Correlation testing:

```
$SPECROOT/RestfulExamples/Events/CreateEventByModelHandleList.xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
This sample event request will create an event of type
0x10f06 (generates a High Memory Utilization alarm).
-->
<rs:event-request throttlesize="10"
xmlns:rs="http://www.ca.com/spectrum/restful/schema/request"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ca.com/spectrum/restful/schema/request
../../../../xsd/Request.xsd">
<rs:event>
<!-- target model of event -->
<rs:target-models>
<rs:model mh="0x100000"/>
</rs:target-models>
<!-- event ID -->
<rs:event-type id="0x10f06"/>
<!-- attributes/varbinds -->
<rs:varbind id="0">75</rs:varbind>
<rs:varbind id="1">99</rs:varbind>
<rs:varbind id="3">mem_instance</rs:varbind>
<rs:varbind id="5">ModelName</rs:varbind>
</rs:event>
</rs:event-request>
```

Note: Replace the 0x100000 in the modelmh field with the correct model handle (while retaining the double quotes).

You can also edit the attribute varbinds to reflect the number of varbinds (or event variables), their index, and value. This example specifies the following four varbinds:

- Event Variable 0: Value = 75 (memory threshold)
- Event Variable 1: Value = 99 (actual memory utilization)
- Event Variable 3: Value = mem_instance (Memory Instance)
- Event Variable 5: Value = name (Memory Instance Name)

Other events can have a different number of varbinds. But this XML example can be edited as appropriate to have the correct number of varbinds.

Configure WizTools RESTClient

Configure the WizTools REST client to work in a CA Spectrum environment. The following steps illustrate how the REST client can be configured before sending the XML request to the OneClick web server.

Note: Any REST client can be used to interact with the CA Spectrum web services. However, in this document, we describe how to configure a specific REST client application. We selected this particular client for its simplicity and ease of use.

Follow these steps:

1. Add the following string to the URL dialog:
`http://OneClick web server hostname/spectrum/restful/events`
2. Click the Method tab and select Post.
3. Click the Body tab and select String Body from the list.
The Body Content-type dialog opens.
4. Supply 'application/xml' for the Content Type.
5. Leave the Charset at 'UTF-8'.
6. Paste the XML contents into the String Body field.
7. Click the Auth tab and select BASIC from the list.
8. Supply values for the following parameters:

Host

Specifies the hostname of the OneClick web server.

Realm

Identifies the type of authentication.

Note: Leave this field blank.

Username

Indicates the username of the operator who is authorized to access OneClick.

Password

Indicates the password.

9. Click Go (>>).
The WizTools REST client is configured.

Create and View Simulated Alarms: An Example

The simulated alarms that you create using the CA Spectrum CLI or web services can be viewed in the model Alarm View. Create the appropriate events on a managed entity, where CA Spectrum simulates that an alarm condition exists (although no actual condition exists). CA Spectrum treats the simulated alarm as if it were an actual alarm on the managed entity. The same intelligence is executed, and the same alarms are displayed in OneClick.

For example, the following image illustrates a simulated Device Contact Lost alarm:

The screenshot displays the CA Spectrum web interface for a simulated alarm. The top section shows the alarm details for device **cis7204-96.5** of type **Cisco7204VXR**. The alarm is titled **DEVICE HAS STOPPED RESPONDING TO POLLS** and is categorized as **DEVICES HAS STOPPED RESPONDING TO POLLS**. The severity is **Critical**, and the date/time is **Apr 5, 2013 12:28:49 PM EDT**. The alarm type is **DEVICES HAS STOPPED RESPONDING TO POLLS**, and the model type is **Rtr_Cisco** with a cause code of **0x10009**.

The **Component Detail** section provides further information about the device and the alarm. It includes a description: **DEVICE HAS STOPPED RESPONDING TO POLLS** and **Device cis7204-96.5 of type Rtr_Cisco has stopped responding to polls and/or external requests. An alarm will be generated.**

The **Severity** is **Critical**, **Impact** is **0**, **Acknowledged** is **No**, **Clearable** is **No**, **Trouble Ticket ID** is **set**, **Assignment** is **shemi11-win7 (0x43200000)**, **Status** is **set**, and **Web Context URL** is **set**.

The **Symptoms** section lists the following: **Device has stopped responding to polls.**

The **Probable Cause** section lists the following: **1) Device Hardware Failure.**, **2) Cable between this and upstream device broken.**, **3) Power Failure.**, **4) Incorrect Network Address.**, **5) Device Firmware Failure.**

The **Actions** section lists the following: **1) Check power to device.**, **2) Verify status lights on device.**, **3) Verify reception of packets.**, **4) Verify network address in device and SPECTRUM.**, **5) Cycle power on device and recheck.**, **6) If above fails, call repair.**

The ModuleOffline event that is created causes the ContactLost alarm to be hidden.

The following image illustrates the ModuleOffline event:

Contents: cis7204-96.5 of type Cisco7204VXR

Alarms Topology List Events Information

Filtered By: Severity

Severity	Date/Time	Name	Type	Alarm Title	Alarm Type	Model Type N...	Cause Code
Critical	Apr 5, 2013 12:30:45 PM EDT	cis7204-96.5	Cisco7204VXR	MODULE OFFLINE DETECTED	MODULE OFFLINE DETECTED	Rtr Cisco	0x10F87

Displaying 1 of 1

Component Detail: cis7204-96.5 of type Cisco7204VXR

Alarm Details Information Impact Host Configuration Root Cause Interfaces Performance Alarm History Neighbors Events Both View

MODULE OFFLINE DETECTED
Apr 5, 2013 12:30:45 PM EDT
This module is offline.

Severity Critical
Impact 0
Acknowledged set
Clearable No
Trouble Ticket ID set
Assignment
Landscape shem11-win7 (0x43200000)
Status set
Web Context URL

Symptoms This module has reported a condition of 'offline'.
Probable Cause This module is offline.
Actions 1) Refer to the Event Message associated with this alarm for additional details that the device may have provided about...
2) Review the Events associated with this model that occurred in the same time frame as this alarm in order to gain insi...

If the Alarm View alarm filter state is changed from its default setting to Show Symptoms, the following alarms are displayed.

- Symptom Alarm: Contact Lost
- Root-Cause Alarm: ModuleOffline

The following image illustrates the Symptom and Root-Cause alarms:

Contents: cis7204-96.5 of type Cisco7204VXR

Alarms Topology List Events Information

Filtered By: Severity

Severity	Date/Time	Name	Type	Alarm Title	Alarm Type	Model Type N...	Cause Code
Critical	Apr 5, 2013 12:28:49 PM EDT	cis7204-96.5	Cisco7204VXR	DEVICE HAS STOPPED RESPONDING TO POLLS	DEVICE HAS STOPPED RESPONDING TO POLLS	Rtr_Cisco	0x10009
Critical	Apr 5, 2013 12:30:45 PM EDT	cis7204-96.5	Cisco7204VXR	MODULE OFFLINE DETECTED	MODULE OFFLINE DETECTED	Rtr_Cisco	0x10f87

Displaying 2 of 2

Component Detail: cis7204-96.5 of type Cisco7204VXR

Alarm Details Information Impact Host Configuration Root Cause Interfaces Performance Alarm History Neighbors Events Path View

DEVICE HAS STOPPED RESPONDING TO POLLS
Apr 5, 2013 12:28:49 PM EDT
Device cis7204-96.5 of type Rtr_Cisco has stopped responding to polls and/or external requests. An alarm will be generated.

Severity Critical
Impact 0
Acknowledged [set](#)
Clearable No
Trouble Ticket ID [set](#)
Assignment
Landscape shemil1-win7 (0x43200000)
Status [set](#)
Web Context URL

Symptoms Device has stopped responding to polls.
Probable Cause
1) Device Hardware Failure.
2) Cable between this and upstream device broken.
3) Power Failure.
4) Incorrect Network Address.
5) Device Firmware Failure.
Actions
1) Check power to device.
2) Verify status lights on device.
3) Verify reception of packets.
4) Verify network address in device and SPECTRUM.

You can clear these alarms by creating the following Clear events:

- ModuleOffline Clear: Event 0x00010f89
- ContactLost Clear: Event 0x00010d30

Index

A

- advanced correlation
 - creating • 17
 - data type comparisons • 68

C

- Caused By, rule relationship • 23
- clear event • 10
- Condition Correlation Editor, opening • 13
- condition parameters
 - creating • 18
 - defined • 18
 - editing • 19
- conditions
 - creating • 17
 - editing • 20

D

- domain
 - adding resources to • 32
 - domains, correlation • 31
 - editing • 33
 - removing resources from • 33

E

- export • 14

F

- fault isolation, contact lost condition • 67

I

- Implied Cause, rule relationship • 11, 23
- Implies, rule relationship • 23
- import • 14

M

- Model Active condition, transfer rules • 67

P

- policies
 - creating • 29
 - editing • 29
 - rules • 29

R

- root cause condition • 23
- root cause target, implies and implied cause relationships • 23
- rules
 - creating • 23
 - criteria in advanced correlation • 23
 - editing • 26
 - relationships • 23

S

- set event • 10

U

- Use as discriminator • 17

V

- varbind variable number • 17