

Sterling Control Center™

How-To Guide

Version 5.2

Sterling Control Center How-To Guide
Version 5.2

First Edition

Copyright © 2003-2010

Sterling Commerce, Inc. ALL RIGHTS RESERVED

Copyright © 2003-2010 Sterling Commerce, Inc. All rights reserved. Additional copyright information is located at the end of the release notes.

STERLING COMMERCE SOFTWARE

*****TRADE SECRET NOTICE*****

THE CONTROL CENTER SOFTWARE (“STERLING COMMERCE SOFTWARE”) IS THE CONFIDENTIAL AND TRADE SECRET PROPERTY OF STERLING COMMERCE, INC., ITS AFFILIATED COMPANIES OR ITS OR THEIR LICENSORS, AND IS PROVIDED UNDER THE TERMS OF A LICENSE AGREEMENT. NO DUPLICATION OR DISCLOSURE WITHOUT PRIOR WRITTEN PERMISSION. RESTRICTED RIGHTS.

This documentation, the Sterling Commerce Software it describes, and the information and know-how they contain constitute the proprietary, confidential and valuable trade secret information of Sterling Commerce, Inc., its affiliated companies or its or their licensors, and may not be used for any unauthorized purpose, or disclosed to others without the prior written permission of the applicable Sterling Commerce entity. This documentation and the Sterling Commerce Software that it describes have been provided pursuant to a license agreement that contains prohibitions against and/or restrictions on their copying, modification and use. Duplication, in whole or in part, if and when permitted, shall bear this notice and the Sterling Commerce, Inc. copyright notice. As and when provided to any governmental entity, government contractor or subcontractor subject to the FARs, this documentation is provided with RESTRICTED RIGHTS under Title 48 52.227-19. Further, as and when provided to any governmental entity, government contractor or subcontractor subject to DFARs, this documentation and the Sterling Commerce Software it describes are provided pursuant to the customary Sterling Commerce license, as described in Title 48 CFR 227-7202 with respect to commercial software and commercial software documentation.

These terms of use shall be governed by the laws of the State of Ohio, USA, without regard to its conflict of laws provisions. If you are accessing the Sterling Commerce Software under an executed agreement, then nothing in these terms and conditions supersedes or modifies the executed agreement.

Where any of the Sterling Commerce Software or Third Party Software is used, duplicated or disclosed by or to the United States government or a government contractor or subcontractor, it is provided with RESTRICTED RIGHTS as defined in Title 48 CFR 52.227-19 and is subject to the following: Title 48 CFR 2.101, 52.227-19, 227.7201 through 227.7202-4, FAR 52.227-14, and FAR 52.227-19(c)(1-2) and (6/87), and where applicable, the customary Sterling Commerce license, as described in Title 48 CFR 227-7202 with respect to commercial software and commercial software documentation including DFAR 252.227-7013, DFAR 252,227-7014, DFAR 252.227-7015 and DFAR 252.227-7018, all as applicable.

The Sterling Commerce Software and the related documentation are licensed either “AS IS” or with a limited warranty, as described in the Sterling Commerce license agreement. Other than any limited warranties provided, NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE. The applicable Sterling Commerce entity reserves the right to revise this publication from time to time and to make changes in the content hereof without the obligation to notify any person or entity of such revisions or changes.

Connect:Direct is a registered trademark of Sterling Commerce. Connect:Enterprise is a registered trademark of Sterling Commerce, U.S. Patent Number 5,734,820. All Third Party Software names are trademarks or registered trademarks of their respective companies. All other brand or product names are trademarks or registered trademarks of their respective companies.

Sterling Commerce, Inc.
4600 Lakehurst Court Dublin, OH 43016-2000 *
614/793-7000

Contents

Chapter 1 Set Up Data Visibility Groups	7
When Should a Data Visibility Group Be Used?	8
How Can I Allow a User to See Only Inventory Processes Across Multiple Servers?	16
How Can I Allow a User to only see data from a specified Remote Node?	17
How Can I Allow a User to Only See When a Specific File Transfer Occurs?	18
How Can I Create a Rule that Only Applies to Data That a Specific User Can See?	19
Chapter 2 Set Object Permissions	21
Can I Control How Users Work With Control Center Objects?	22
How Can I Let Everyone Use a Calendar But Only Allow an Administrator to Update the Calendar?	23
How Can I Restrict the Use and Update of a Calendar to Limited Users?	24
How Can I Allow All Users to Use a Calendar But Only One User to Update the Calendar?	25
Chapter 3 Set Up Actions	27
How Can I Create an Action to Release a Connect:Direct Process?	28
How Can I Create an Action that Sends an E-mail that Includes an SLC Name?	29
How Can I Create an Action to Send an E-mail and Generate an Alert?	31
How Do I Generate a High-Severity Alert and an SNMP Trap?	33
How Do I Pass an Operating System Command and Variables to Open a Help Desk Ticket?	35
Chapter 4 Set Up Rules	37
How Can I Know When a Process Completes?	39
How Can I Know When a Process Has a Problem?	40
How Can I Know When a Copy Completes?	41
How Can I Know When a Copy Completes Unsuccessfully?	42
How Can I Know When a Monitored Server Goes Down?	43
How Can I Know When a Monitored Server Starts?	44

How Can I Send an E-Mail in the Event of a Critical Database Problem?	45
How Can I Generate a Server Down Alert Only After the Server Has Been Down More Than Five Minutes?	46
How Can I Generate an Alert When a Managed Server Stops and Handle t he Alert Automatically When the Server Restarts?	48
How Can I Know When a Connect:Direct Process Goes in the Held in Error Queue?	50
How Can I Know When a Connect:Enterprise for UNIX Daemon Stops?	51
How Can I Know When a Connect:Enterprise for UNIX Daemon Starts?	52
How Can I Know When a Connect:Enterprise Daemon Goes Down and Stays Down for More Than Five Minutes?.	53
How Can I Know When a Sterling Integrator Adapter has Stopped?	54
How Can I Know When a Sterling Integrator Adapter is Enabled?	55
How Can I Know When a Sterling Integrator Adapter Goes Down and Stays Down for More Than Five Minutes?.	56
How Can I Know When a Connect:Enterprise Remote Connection Request Operation was Performed?	57
How Can I Know When a Connect:Enterprise Remote Connection Add Operation was Performed?	58
How Can I Know When a Connect:Enterprise for UNIX Offline Add was Performed?	59
How Can I Know When a Connect:Enterprise for UNIX AutoConnect Fails?	60
How Can I Know When a Connect:Enterprise Remote Connect Connection is Performed?	61
How Can I Know When a Connect:Enterprise Remote Connect Connection is Performed?	62
How Can I Know When a Connect:Enterprise Remote Connect Delete is Performed?	63
How Can I Know When a Connect:Enterprise Remote Connect Directory List is Performed?	64
How Can I Know When a Connect:Enterprise Remote Connect Disconnection is Performed?	65
How Can I Know When a Connect:Enterprise for UNIX Offline Extraction is Performed?	66
How Can I Know When a Connect:Enterprise for z/OS AutoConnect CONNECT Operation is Performed?	67
How Can I Know When a Connect:Enterprise for z/OS AutoConnect DISCON Operation is Performed?	68
How Can I Know When a Connect:Enterprise for z/OS AutoConnect Send Operation is Performed?	69
How Can I Know When a Connect:Enterprise for z/OS AutoConnect Receive Operation is Performed?	70
How Can I Know When a Connect:Enterprise for UNIX AutoConnect Collect Operation is Performed?	71
How Can I Know When a Connect:Enterprise for UNIX AutoConnect Transmit Operation is Performed?	72
How Can I Know When a Managed Server's License is About to Expire?	73
How Can I Avoid Receiving Alerts From Both Servers Involved in a Process?	74
How Can I Specify More Than One Message ID in the Rule Criteria?	75

How Can I Know If a New Version of Objects was Created?	76
How Can I Know When Control Center Is Not Monitoring a Server Regularly?	77
How to Get Notified When a Sterling File Gateway Route Fails?	78
How Can a Producer Know When a File Was Delivered to a Sterling Integrator Mailbox?	79
How Can a Producer Know if a File Was Delivered to a Consumer's Mailbox?	80
How Can a Producer Get Notified if a File Does Not Arrive in Their Mailbox Within 5 Minutes After the Connection is Initiated?	81
Chapter 5 Troubleshooting Rules	83
<hr/>	
Troubleshoot Rules	83
Chapter 6 Set Up Schedules	87
<hr/>	
How Can I Build a Schedule To Prevent Rules From Triggering During My Maintenance Windows?	87
Chapter 7 Reports Example	91
<hr/>	
How Do I Set Up an Automated Report of the Previous Day's Alerts?	91
Chapter 8 Set Up SLCs	93
<hr/>	
What's the Correlation Between SLCs and Rules?	94
When Should an SLC Be Used Versus a Rule?	95
When Do I Need SLCs?	96
What Events May Be Generated By SLCs and When?	97
SLC Notifications for Calendar Schedule-Based SLCs	97
SLC Message IDs Associated with Start Range	97
SLC Message IDs Associated with End Range	97
Message IDs for Calendar Schedule-Based SLCs	98
SLC Notifications for Duration Schedule-Based SLCs	99
Message IDs for Duration Schedule-Based SLCs	99
Other Message IDs for Both Calendar and Duration Schedule-Based SLCs	99
How Do I Know Which Type of SLC to Use?	101
How Can I Be Alerted If One Process Runs Too Long?	102
How Can I Be Alerted If a Critical Process Fails to Complete On Time?	105
How Can I Be Alerted When Related Connect:Enterprise Batches Arrive?	106
How Can I Be Alerted If a Related Group of Processes Runs Too Long?	112
How Can I Be Alerted If a Related Group of Critical Processes Don't All Complete On Time?	118
How Can I Be Alerted If a Critical Process Doesn't Start On Time?	119
How Can I Be Alerted If One Or More Related Processes Do Not Start or Complete On Time?	120
How Can I Be Alerted If One Or More Related Processes May Not Complete On Time?	121
How Can I Reduce the Number of Events with the Same Message Generated By	

Workflow SLCs?	124
How Can I Be Alerted If A Connect:Direct File Transfer May Not Complete On Time?	126
How Can I Be Alerted If an Hourly Process Isn't On Time?	127
How Can a Producer Know When a File Was Not Delivered On Time?	132
How Can a Producer Get Notified if the Consumer Does Not Pick Up a File Within 15 Minutes?	134
How Can I Use RUNBATCH To Create Schedules for an Hourly Process?	138
Template	138
Script	138

Chapter 9 Troubleshoot SLCs **143**

Troubleshoot SLCs	143
SLC Tips	144
How Do I Fix My SLC?	145
SLC Not Enabled	145
Match Criteria	145
A.M. versus P.M.	147
Monitoring Windows	147
Monitored Servers	149
Rules and Actions	150
Milestone Times	152
SLC Order	152
Time Zones	152
Everything's Right Except the Alerts	153
What Else Can I Do?	155

Chapter 10 Update Information for Multiple Servers **169**

How Can I Do a Bulk Update of the Passwords Used by Control Center for Monitored Servers?	169
---	-----

Index **171**

Set Up Data Visibility Groups

This chapter shows how to set up data visibility groups to address common situations.

These examples do not describe the detailed steps necessary to set up data visibility groups. Some examples describe only the idea behind a solution to a problem. For a detailed explanation of data visibility groups and how to set them up, consult the Help system or the *Sterling Control Center System Administration Guide*.

These examples are provided:

- When Should a Data Visibility Group Be Used?
- How Can I Allow a User to See Only Inventory Processes Across Multiple Servers?
- How Can I Allow a User to only see data from a specified Remote Node?
- How Can I Allow a User to Only See When a Specific File Transfer Occurs?
- How Can I Create a Rule that Only Applies to Data That a Specific User Can See?

When Should a Data Visibility Group Be Used?

Data visibility groups should be used when there is a need to limit what data (events) a specific user can monitor.

For example, assume that a company has one Connect:Direct server on which both Payroll and Order Processing work is performed. The Control Center administrator wants to give both Payroll and Order Processing personnel access to Control Center functions, but doesn't want Order Processing personnel to have any visibility to the Payroll file activity; nor the Payroll personnel to have any access to the Order Processing activity.

Data visibility groups in Control Center provide this capability.

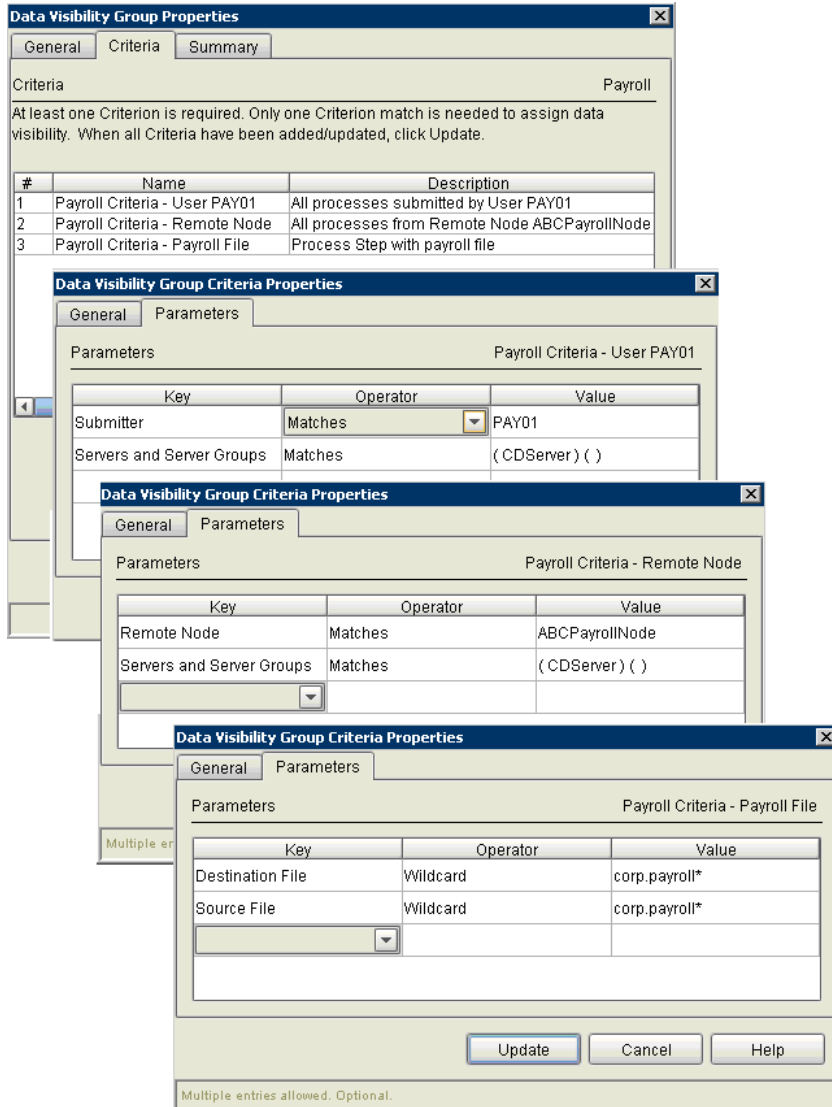
Data visibility groups are basically a set of rules that are applied to events that are created in Control Center and when the rule matches the criteria specified, the name of the Data Visibility Group is added to the attributes of the event. So, the Control Center administrator can set up a Data Visibility Group called Payroll that will classify events as "Payroll" events (by tagging them with the "Payroll" Data Visibility Group attribute) when any one of these criteria is met:

- Any Process is submitted by the user PAY01 is a Payroll process.
- Any Process that runs between this Connect:Direct server and the remote Payroll server is a Payroll process.
- Any Process Step that deals with a file that contains "corp.payroll" in the name is a Payroll Process Step.

And the Control Center administrator can set up a data visibility group called OrderProcessing that will classify events as "OrderProcessing" events when any one of these criteria is met:

- All Processes submitted by user ORD01 are Order Processing processes.
- All Processes submitted by user ORD02 are Order Processing processes.
- All Process Steps that deal with files that begin with "ORDER." are Order Processing processes.

The following shows a sample Payroll data visibility group:



The following shows a sample Order Processing data visibility group:

The screenshot displays a 'Data Visibility Group Properties' dialog box with three overlapping 'Data Visibility Group Criteria Properties' sub-dialogs. The main dialog has tabs for 'General', 'Criteria', and 'Summary'. The 'Criteria' tab is active, showing a list of criteria:

#	Name	Description
1	OrderProcessing Criteria 1 - User ORD01	Processes submitted by User ORD01
2	OrderProcessing Criteria 2 - User ORD02	Processes submitted by User ORD01
3	OrderProcessing Criteria 3 - File ORDER	All processes using file ORDER*

The 'OrderProcessing Criteria 1 - User ORD01' sub-dialog shows parameters:

Key	Operator	Value
Submitter	Matches	ORD01*
Servers and Server Groups	Matches	(CDServer) ()

The 'OrderProcessing Criteria 2 - User ORD02' sub-dialog shows parameters:

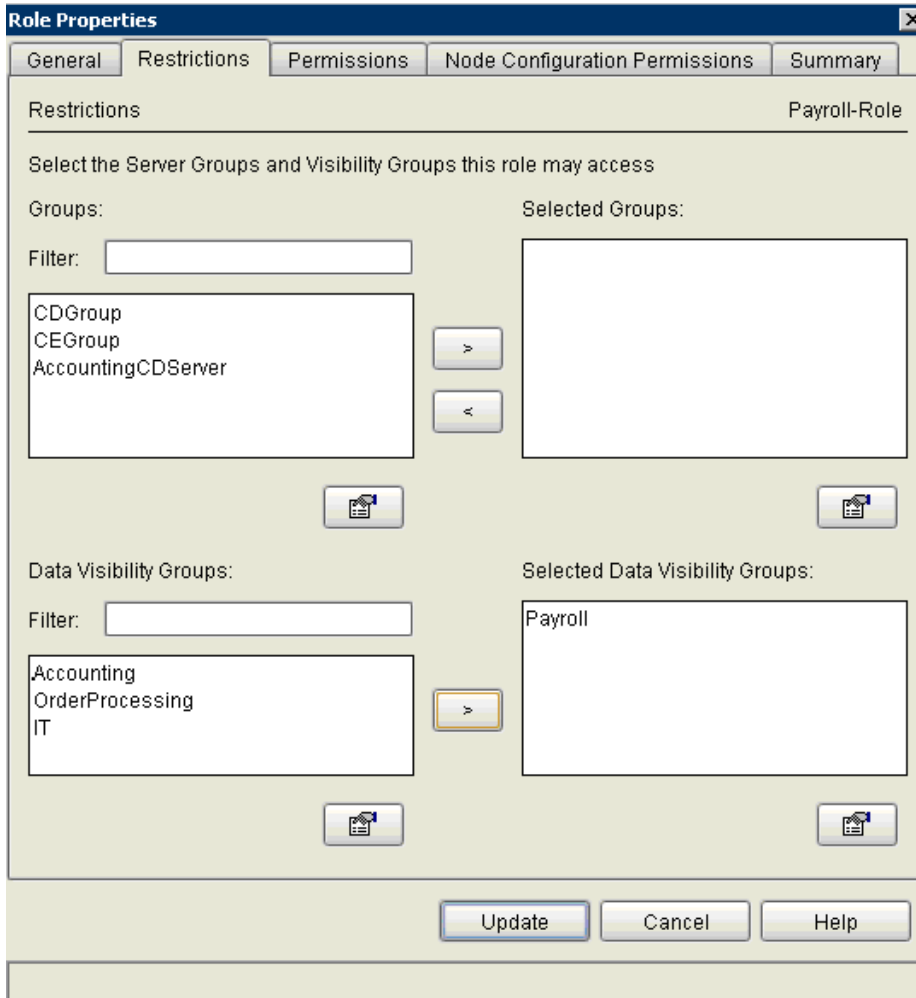
Key	Operator	Value
Submitter	Matches	ORD02
Servers and Server Groups	Matches	(CDServer) ()

The 'OrderProcessing Criteria 3 - File ORDER' sub-dialog shows parameters:

Key	Operator	Value
Destination File	Matches	ORDER*
Source File	Matches	ORDER*
Servers and Server Groups	Matches	(CDServer) ()

Buttons for 'Update', 'Cancel', and 'Help' are visible at the bottom of the sub-dialogs. A note at the bottom of the main dialog states: 'Multiple entries allowed. Optional.'

A Control Center role can now be defined that is restricted to data visibility group Payroll.



Another role can be defined that is restricted to data visibility group OrderProcessing.

The screenshot shows the 'Role Properties' dialog box with the 'Restrictions' tab selected. The title bar reads 'Role Properties' and the window title is 'OrderProcessing - Role'. The dialog is divided into two main sections: 'Groups' and 'Data Visibility Groups'. Each section has a 'Filter' input field, a list of available items, and a list of selected items. In the 'Groups' section, the available list contains 'CDGroup', 'CEGroup', and 'AccountingCDServer', and the selected list is empty. In the 'Data Visibility Groups' section, the available list contains 'Accounting', 'IT', and 'Payroll', and the selected list contains 'OrderProcessing'. Navigation buttons (right arrow, left arrow) are present between the lists. At the bottom of the dialog are 'Update', 'Cancel', and 'Help' buttons.

Section	Filter	Available Items	Selected Items
Groups		CDGroup CEGroup AccountingCDServer	
Data Visibility Groups		Accounting IT Payroll	OrderProcessing

Control Center users can now be defined and assigned to the appropriate Payroll or Order Processing roles.

The screenshot shows a 'User Properties' dialog box with the following fields and values:

- User ID: PayrollUser
- Password: [Redacted]
- Re-type Password: [Redacted]
- Host Name: [Empty]
- Windows Domain: [Empty]
- TCP/IP Address: [Empty]
- Description: [Empty]
- Role: Payroll - Role

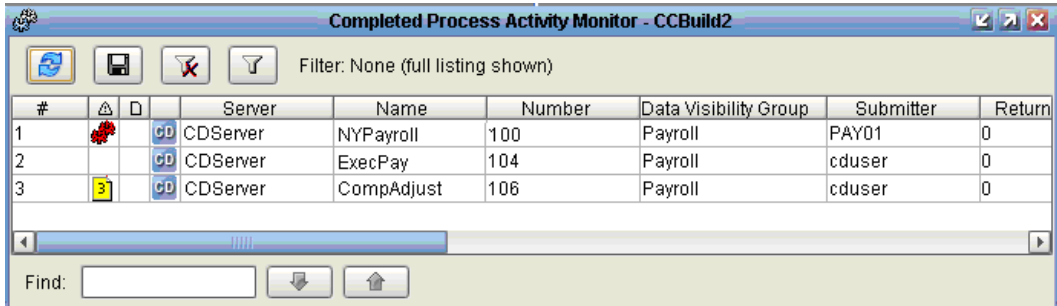
Buttons at the bottom: Update, Close, Help.

The screenshot shows a 'User Properties' dialog box with the following fields and values:

- User ID: OrderProcessingUser
- Password: [Redacted]
- Re-type Password: [Redacted]
- Host Name: [Empty]
- Windows Domain: [Empty]
- TCP/IP Address: [Empty]
- Description: [Empty]
- Role: OrderProcessing - Role

Buttons at the bottom: Update, Close, Help.

User PayrollUser could then sign on to the Control Center console and would only see Payroll activities as if those activities were the only activities in the Connect:Direct server.

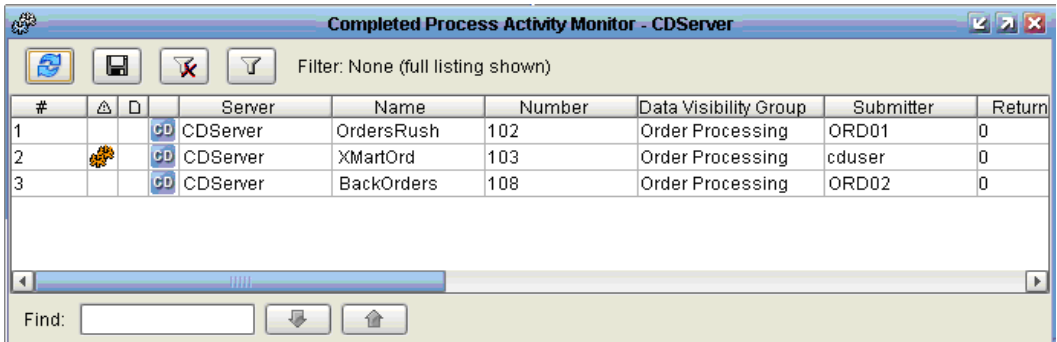


The screenshot shows a window titled "Completed Process Activity Monitor - CCBUILD2". It features a toolbar with icons for refresh, save, delete, and filter. Below the toolbar, it says "Filter: None (full listing shown)". The main area contains a table with the following data:

#			Server	Name	Number	Data Visibility Group	Submitter	Return
1			CD CDServer	NYPayroll	100	Payroll	PAY01	0
2			CD CDServer	ExecPay	104	Payroll	cduser	0
3			CD CDServer	CompAdjust	106	Payroll	cduser	0

At the bottom, there is a "Find:" text box and two arrow buttons.

And User OrderProcessingUser would only see Order Processing activities.

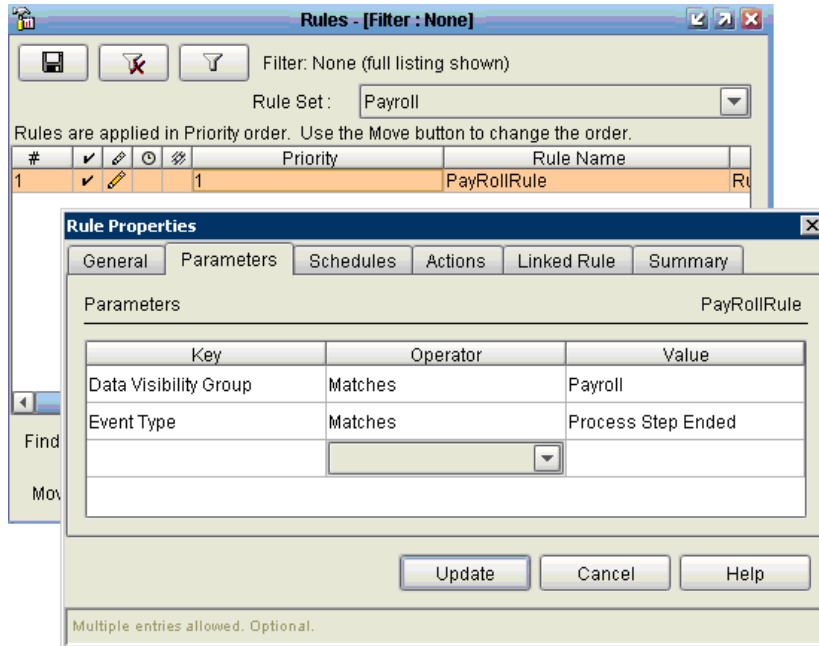


The screenshot shows a window titled "Completed Process Activity Monitor - CDSERVER". It features a toolbar with icons for refresh, save, delete, and filter. Below the toolbar, it says "Filter: None (full listing shown)". The main area contains a table with the following data:

#			Server	Name	Number	Data Visibility Group	Submitter	Return
1			CD CDServer	OrdersRush	102	Order Processing	ORD01	0
2			CD CDServer	XMartOrd	103	Order Processing	cduser	0
3			CD CDServer	BackOrders	108	Order Processing	ORD02	0

At the bottom, there is a "Find:" text box and two arrow buttons.

If the Payroll role were permitted to define rules or SLCs, those rules or SLCs would only apply to Payroll activity and would have no effect on Order Processing activity.



How Can I Allow a User to See Only Inventory Processes Across Multiple Servers?

Create an Inventory data visibility group with a criterion that includes selections of Servers and a Process Name that represents inventory processes. Create a role that is restricted to the Inventory data visibility group and assign that role to the user who is only allowed to see inventory processes.

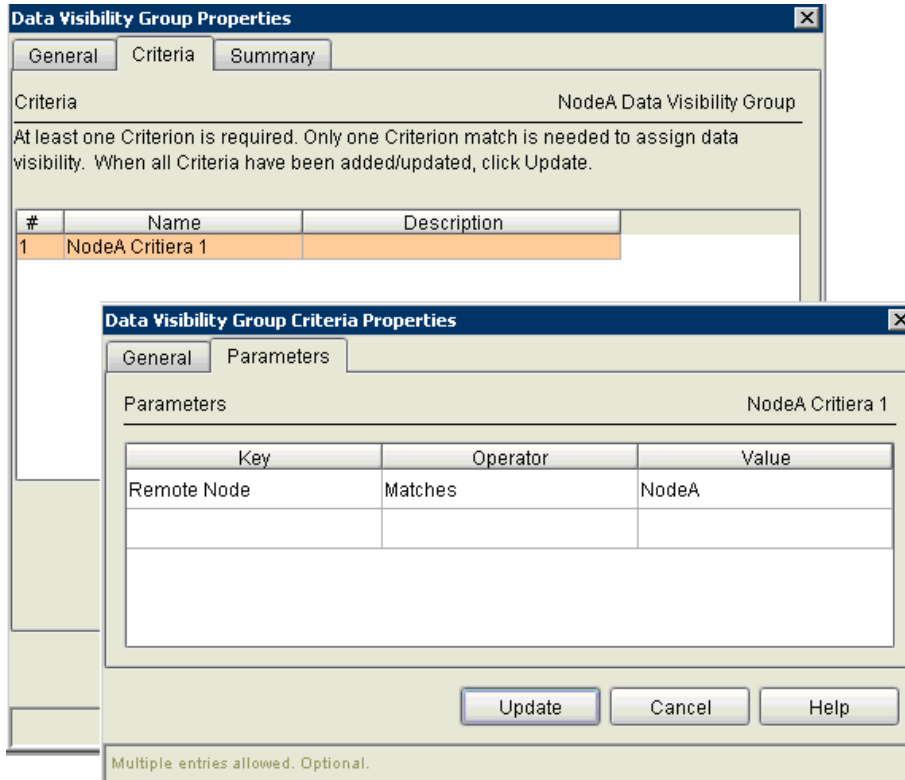
The screenshot shows a dialog box titled "Create Data Visibility Group Criteria" with a "Parameters" tab selected. The dialog contains a table with the following data:

Key	Operator	Value
Servers and Server Group	Matches	(CDServer CEzOS) ()
Process Name	Matches	Inventory*

At the bottom of the dialog, there are buttons for "< Back", "Next >", "Finish", "Cancel", and "Help". The "Finish" button is highlighted. A footer note states "Multiple entries allowed, Optional." The Sterling Commerce logo is visible in the bottom left corner of the dialog.

How Can I Allow a User to only see data from a specified Remote Node?

Create a data visibility group with a criterion that specifies the Remote Node.



Then create a role with the data visibility group selected and assign that role to your user.

How Can I Allow a User to Only See When a Specific File Transfer Occurs?

Create a data visibility group with a criterion that defines the specific file transfer that the user should be able to see. Create a role that is restricted to that data visibility group and assign that role to the user.

The image shows two overlapping dialog boxes. The background dialog is titled "Data Visibility Group Properties" and has tabs for "General", "Criteria", and "Summary". The "Criteria" tab is active, showing a text field with "FileABC" and a message: "At least one Criterion is required. Only one Criterion match is needed to assign data visibility. When all Criteria have been added/updated, click Update." Below this is a table with one entry:

#	Name	Description
1	FileABC Criteria	

The foreground dialog is titled "Data Visibility Group Criteria Properties" and has tabs for "General" and "Parameters". The "Parameters" tab is active, showing a text field with "FileABC Criteria" and a table with the following data:

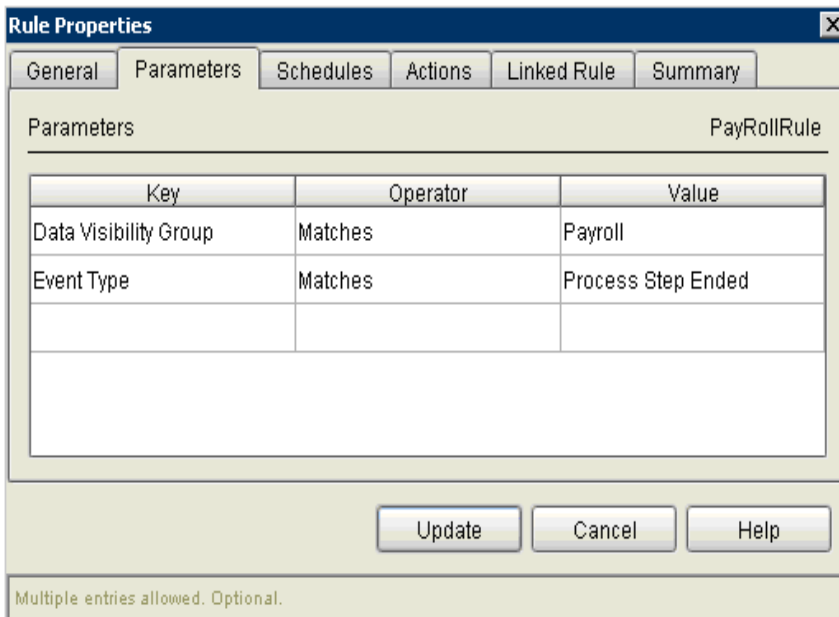
Key	Operator	Value
Destination File	Matches	FileABC
From Server	Matches	FTPServerXYZ

At the bottom of the foreground dialog are buttons for "Update", "Cancel", and "Help". A footer note reads "Multiple entries allowed. Optional."

How Can I Create a Rule that Only Applies to Data That a Specific User Can See?

Create a data visibility group that defines the data you want the user to see. Create a role that is restricted to that data visibility group. Assign that role to the user.

Let the Control Center administrator create the rule and set the data visibility group of the user's role in the rule's parameters. Or, have the user create his own rules; the user will be forced to supply his data visibility group as part of the rule's parameter since he is restricted to that data visibility group.



Rule Properties

General Parameters Schedules Actions Linked Rule Summary

Parameters PayRollRule

Key	Operator	Value
Data Visibility Group	Matches	Payroll
Event Type	Matches	Process Step Ended

Update Cancel Help

Multiple entries allowed. Optional.

Set Object Permissions

This chapter shows how to set up permissible objects to address common situations.

These examples do not describe the detailed steps necessary to set up permissible objects. Some examples describe only the idea behind a solution to a problem. For a detailed explanation of permissible objects and how to set them up, consult the Help system or the Sterling Control Center System Administration Guide.

These examples are provided:

- Can I Control How Users Work With Control Center Objects?
- How Can I Let Everyone Use a Calendar But Only Allow an Administrator to Update the Calendar?
- How Can I Restrict the Use and Update of a Calendar to Limited Users?
- How Can I Allow All Users to Use a Calendar But Only One User to Update the Calendar?

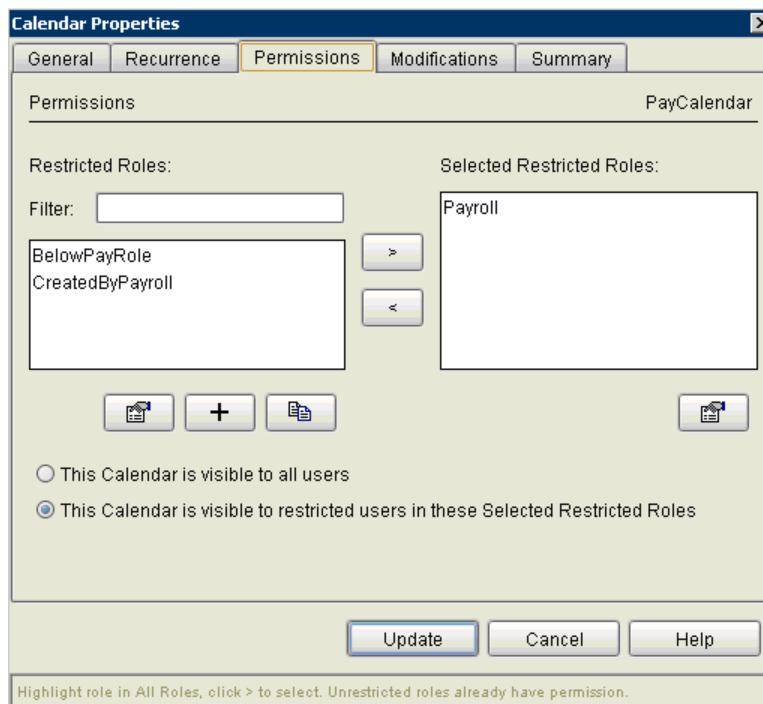
Can I Control How Users Work With Control Center Objects?

You can use object permissions to limit the users who can view/use an object and the users who can update objects. You can control these types of permissions for the following objects in Control Center:

Note: Unrestricted users can always view/use the objects.

- Rule Actions
- Rule Schedules
- SLC Schedules
- Message Lists
- Metadata Schedules
- Calendars
- E-mail Lists
- Report Schedules
- Metadata Actions

You set object permissions on the Permissions tab for the object type you are working with. The following example shows the Permissions tab for a calendar object:

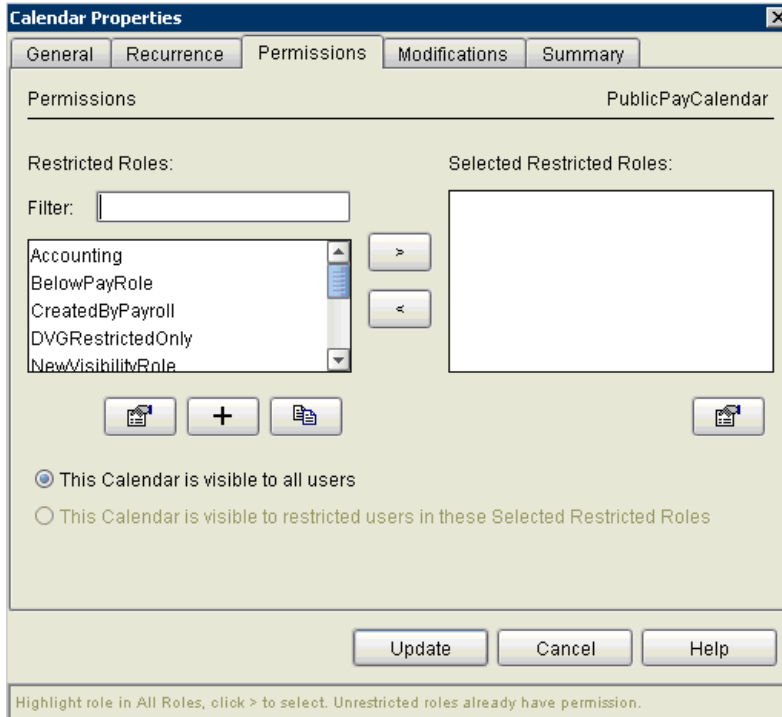


On the Permissions panel:

- “Selected Restricted Roles:” identify which Restricted roles can update the object.
- The radio button identifies the users who can view/use the object

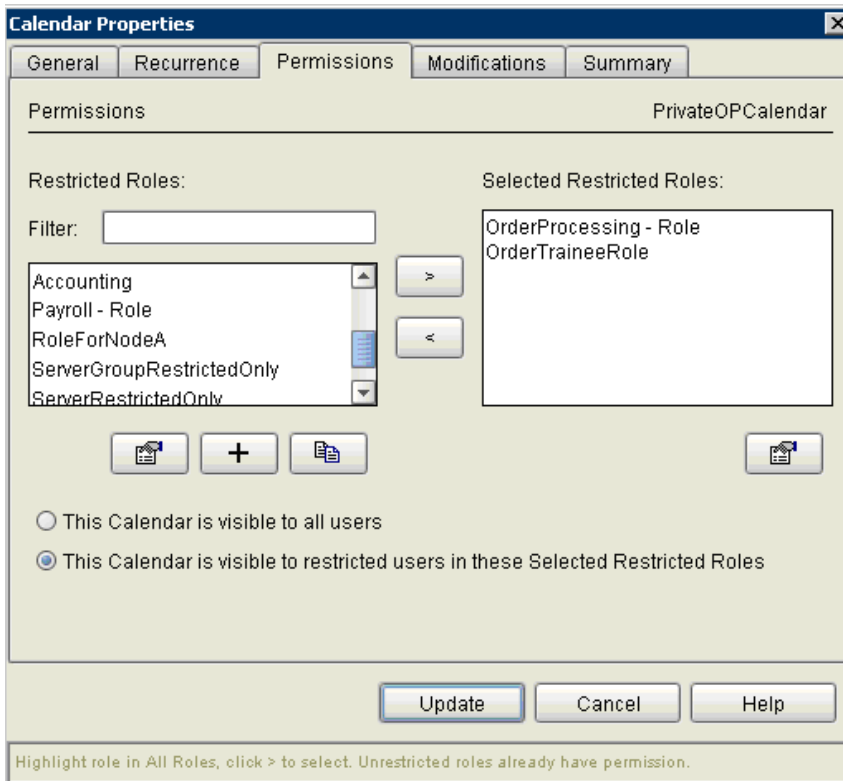
How Can I Let Everyone Use a Calendar But Only Allow an Administrator to Update the Calendar?

Select “This Calendar is visible to all users” on the Calendar’s Permissions tab.



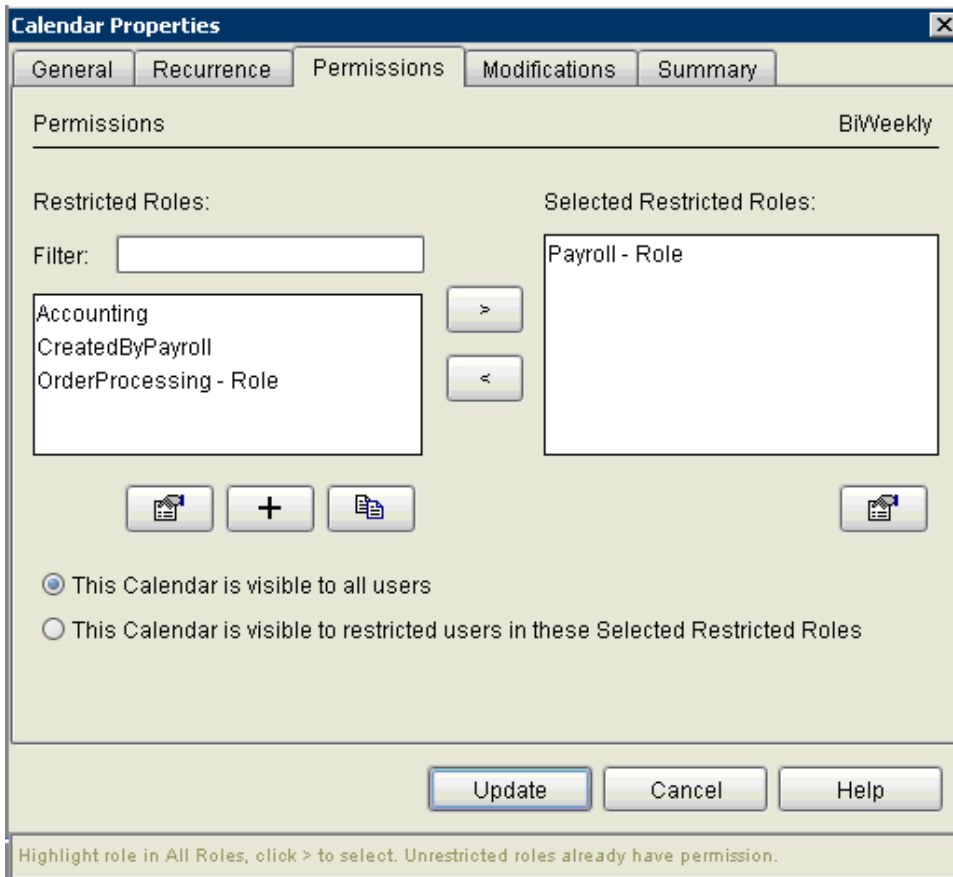
How Can I Restrict the Use and Update of a Calendar to Limited Users?

Select “This Calendar is visible to restricted users in these Selected Restricted Roles” on the calendar's Permissions tab and specify the roles of the users you want to be able to use and update the calendar in the Selected Restricted Roles.



How Can I Allow All Users to Use a Calendar But Only One User to Update the Calendar?

Select “This Calendar is visible to all users” on the calendar's Permissions tab and specify the role of the user you want to be able to update the calendar in the Selected Restricted Roles. Don't assign the role to any other user.



Set Up Actions

This chapter shows how to set up actions to address common situations.

These examples do not describe the detailed steps necessary to set up actions. Some examples describe only the idea behind a solution to a problem. For a detailed explanation of actions and how to set them up, consult the Help system or the *Sterling Control Center System Administration Guide*.

These examples are provided:

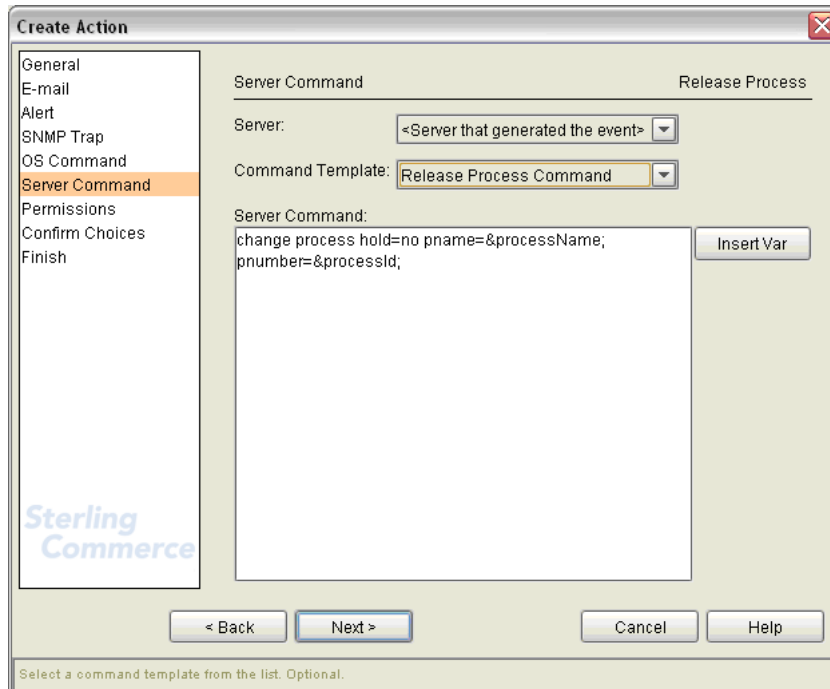
- How Can I Create an Action to Release a Connect:Direct Process?
- How Can I Create an Action that Sends an E-mail that Includes an SLC Name?
- How Can I Create an Action to Send an E-mail and Generate an Alert?
- How Do I Generate a High-Severity Alert and an SNMP Trap?
- How Do I Pass an Operating System Command and Variables to Open a Help Desk Ticket?

How Can I Create an Action to Release a Connect:Direct Process?

Create an action with Server Command equal to the following:

```
"change process hold=no pname=&processName; pnumber=&processid;"
```

You can do this easily using the Release Process Command command template provided in the Create Action wizard.

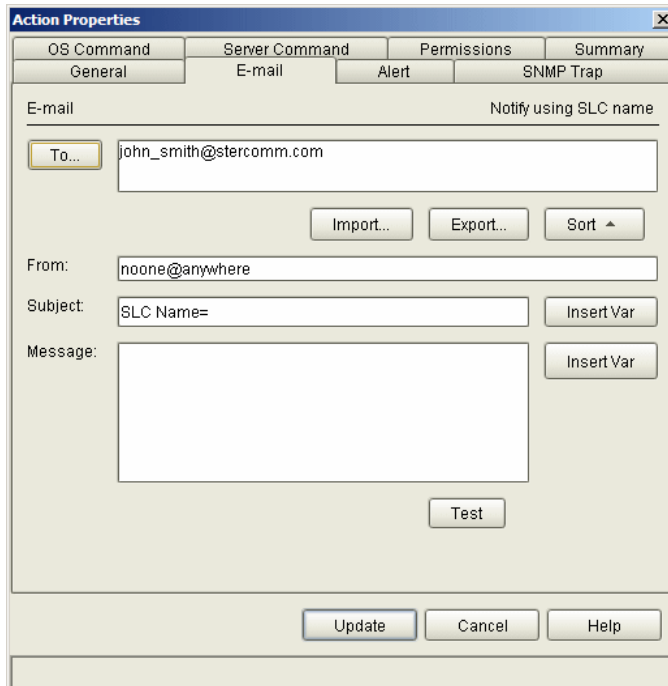


To make this action release Processes to attempt to run again, you could specify this action in a rule that, for example, watches for Connect:Direct Processes that have transitioned to the Held in Error queue. (See *How Can I Know When a Connect:Direct Process Goes in the Held in Error Queue?* on page 50.)

How Can I Create an Action that Sends an E-mail that Includes an SLC Name?

To plug the SLC name in the subject line in e-mails generated as a result of SLC events, you can use the event element variable called `groupId`.

First, you create an action and add the e-mail address along with the descriptive text you want in the Subject line.

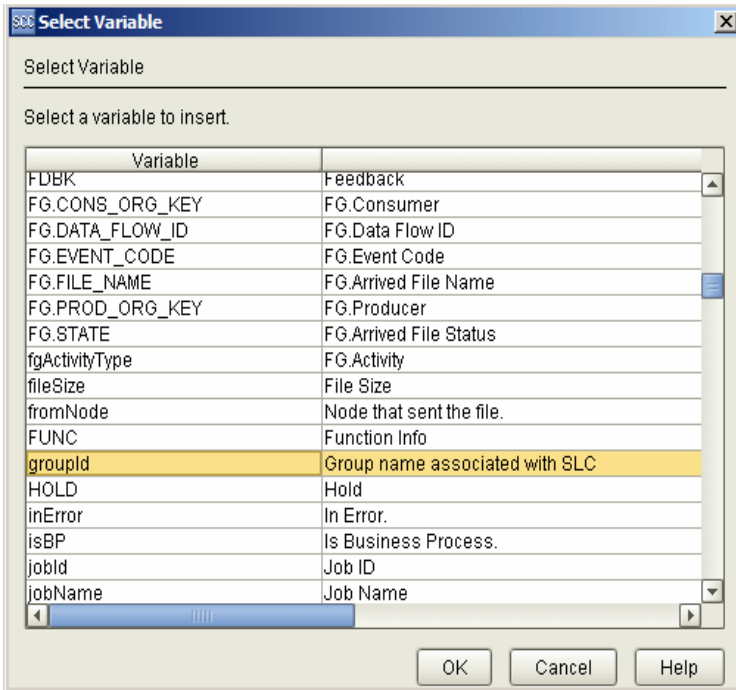


The screenshot shows the "Action Properties" dialog box with the "E-mail" tab selected. The dialog has a title bar with a close button (X). Below the title bar are four tabs: "OS Command", "Server Command", "Permissions", and "Summary". The "E-mail" tab is active, and it contains a sub-tab "E-mail" and a sub-tab "Alert". The "E-mail" sub-tab is selected, and it displays the following fields and buttons:

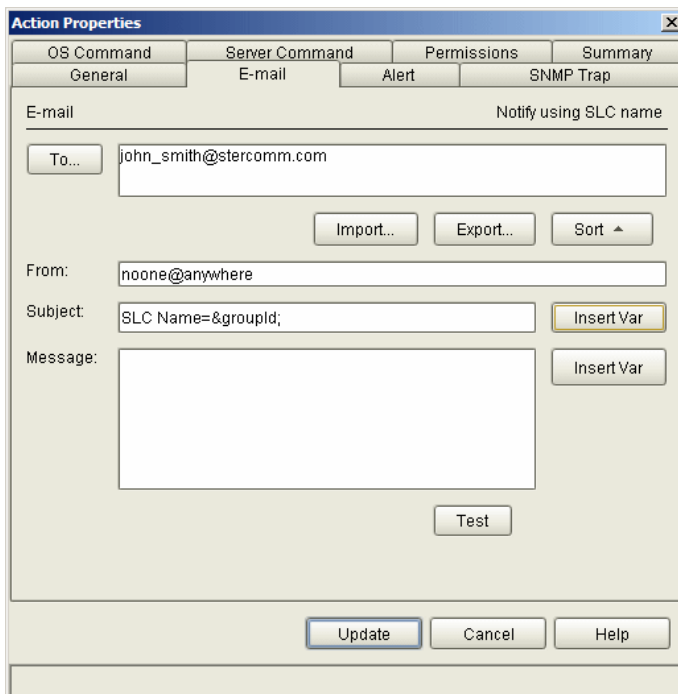
- To...:** A text field containing "john_smith@stercomm.com".
- From:** A text field containing "noone@anywhere".
- Subject:** A text field containing "SLC Name=".
- Message:** A large empty text area.
- Buttons:** "Import...", "Export...", "Sort ▲", "Insert Var" (next to Subject), "Insert Var" (next to Message), "Test", "Update", "Cancel", and "Help".

The "Test" button is located at the bottom center of the dialog. The "Update", "Cancel", and "Help" buttons are located at the bottom right of the dialog.

Click **Insert Var** and select **groupId** in the Select Variable window.



E-mails generated as a result of this action will contain the SLC name in their subject.



How Can I Create an Action to Send an E-mail and Generate an Alert?

You can create an action that sends an e-mail notification to a system administrator and generates an in-compliance alert. Specify this action in a rule set up to trigger when a certain critical Process completes successfully.

The following example illustrates this kind of setup.

Click **Manage > Rules and Actions > Actions** and click + to create a new action.

Complete **Name** and **Description**.

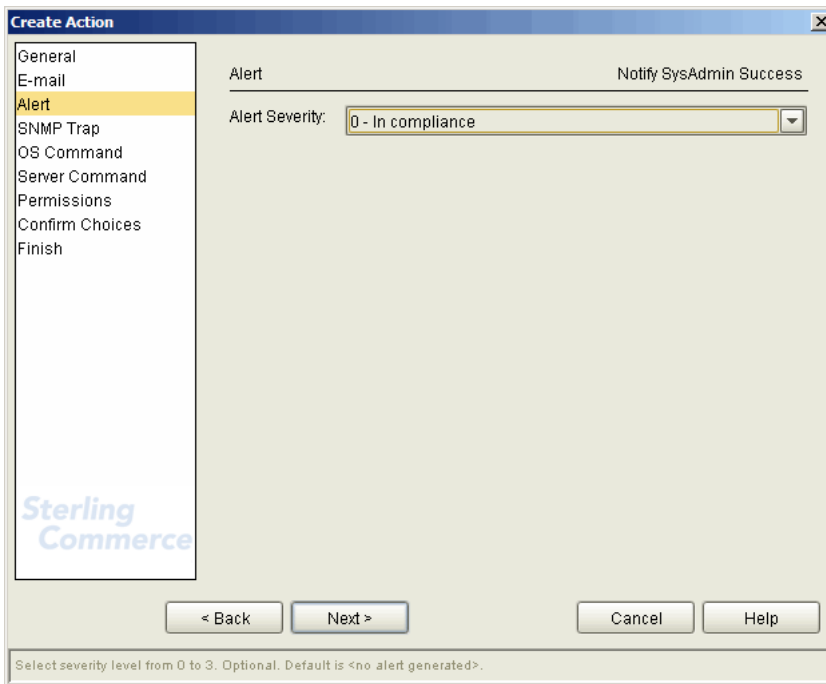
Click **Next** and create e-mail entries similar to the following. Use **Insert Var** to add variables to Subject and Message.

The screenshot shows the 'Create Action' dialog box with the 'E-mail' tab selected. The configuration is as follows:

- Title:** Notify SysAdmin Success
- To:** Admin@Mycompany.com
- From:** SCCConsole@Mycompany.com
- Subject:** &processId; &processName; completed
- Message:** Process &processId; &processName; has completed normally.

Buttons include 'Import...', 'Export...', 'Sort', 'Insert Var', 'Test', '< Back', 'Next >', 'Cancel', and 'Help'. A footer note states: 'Required if To address is specified. Special characters allowed.'

Click **Next** and choose **0 - In compliance** for Alert Severity.



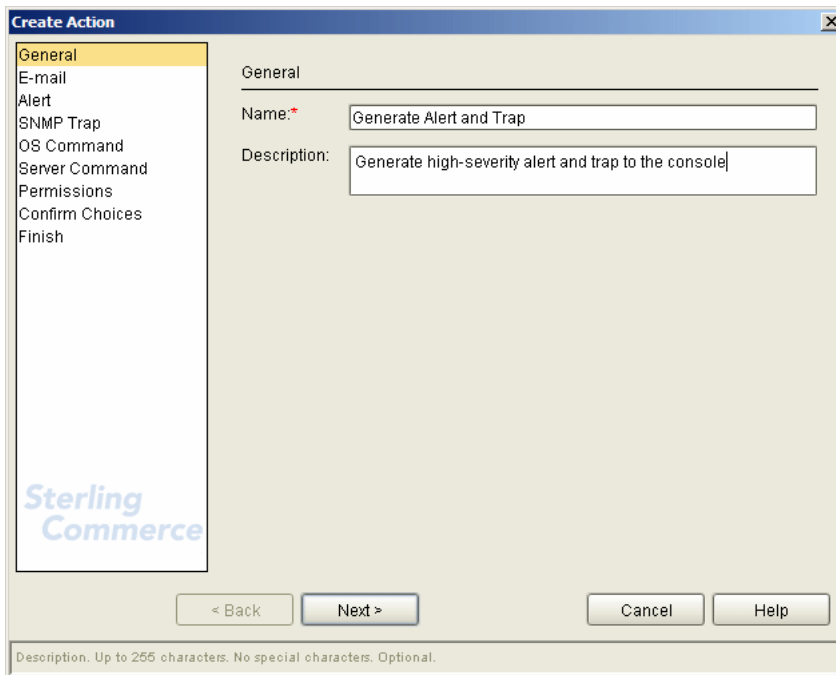
Leave the remaining wizard panels unchanged.

How Do I Generate a High-Severity Alert and an SNMP Trap?

You can set up an action that generates a high-severity alert and an SNMP trap to an Enterprise System Management (ESM) host. The ESM host must be defined in the Sterling Control Center ESM Host settings. (See “Sterling Control Center Settings” in the Help or the *Sterling Control Center System Administration Guide*.)

You could specify this action in a rule that triggers, for example, whenever any file transfer on a managed Connect:Enterprise server fails (that is, completes with a return code greater than 4).

Create an action with a **Name** and **Description** similar to the following.



Create Action

General

E-mail

Alert

SNMP Trap

OS Command

Server Command

Permissions

Confirm Choices

Finish

General

Name:*

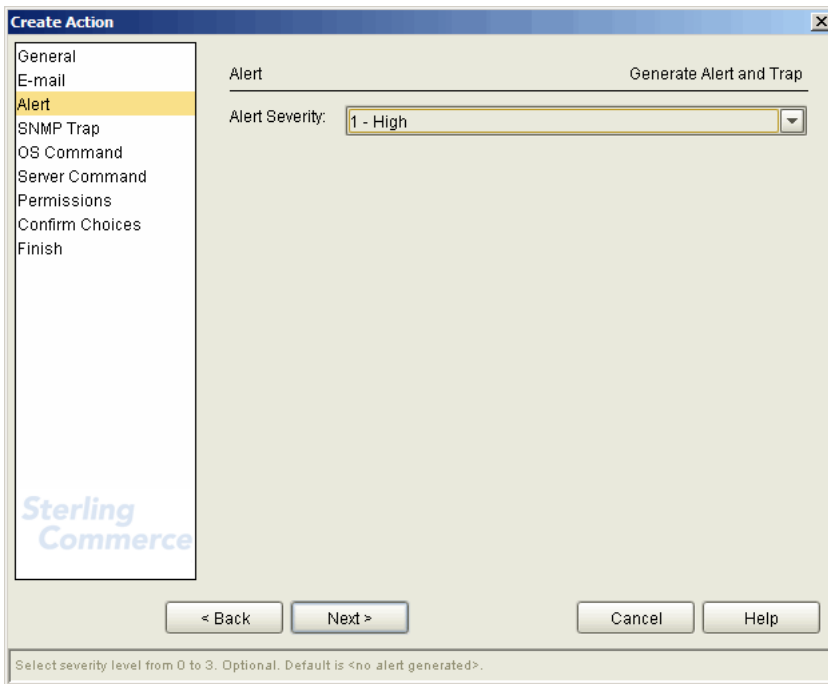
Description:

Sterling Commerce

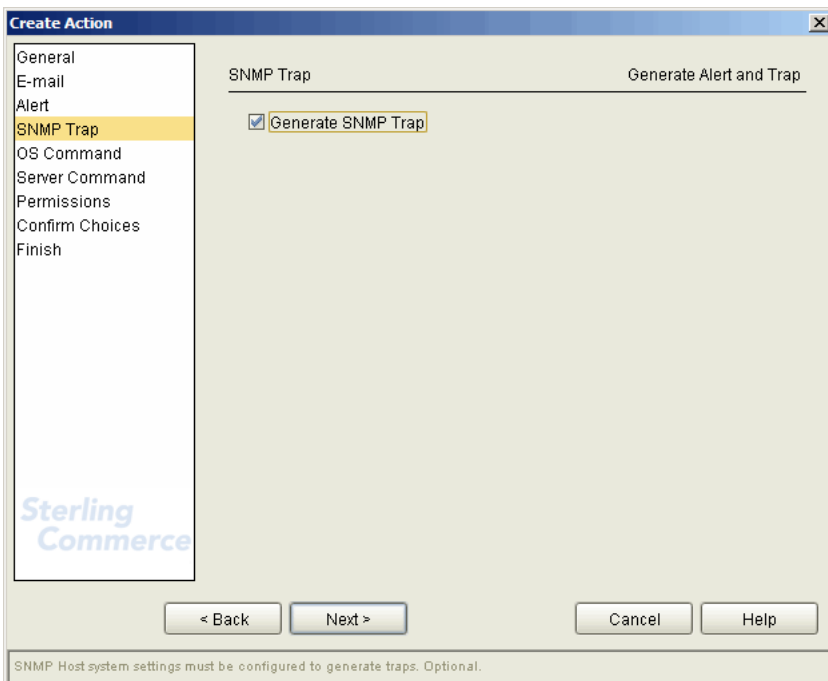
< Back Next > Cancel Help

Description. Up to 255 characters. No special characters. Optional.

Skip the E-mail wizard panel and go to the Alert panel. Select **1 - High** for severity and click **Next**.



On the SNMP Trap panel, select **SNMP Trap**. Then click through subsequent panels and confirm your choices.

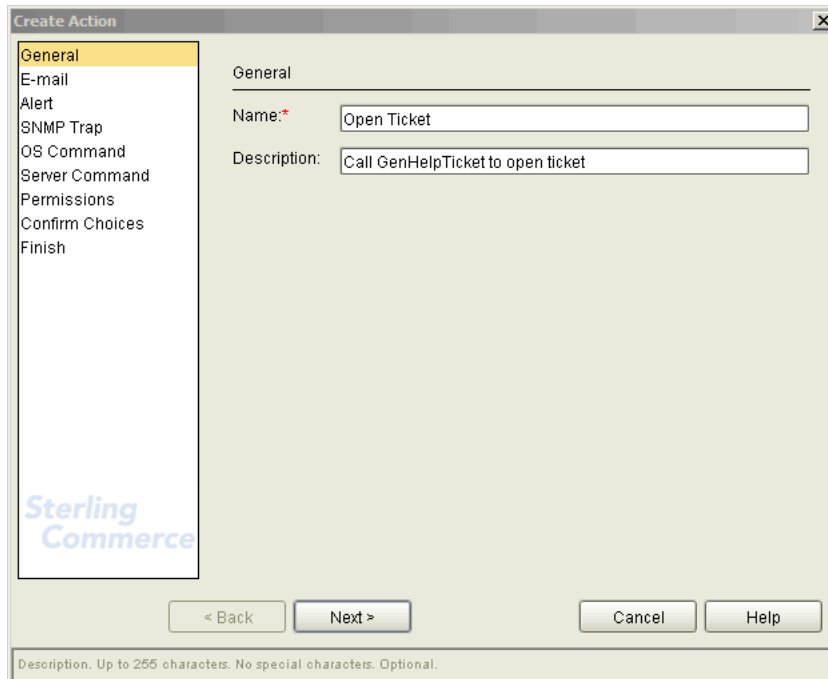


How Do I Pass an Operating System Command and Variables to Open a Help Desk Ticket?

You can create an action that calls a user-written program (named GenHelpTicket.exe in this example) that opens a Help Desk ticket when the rule it is attached to is triggered. The action also passes an XML string of event data to the user-written program.

When calling an operating system command, remember that the called program must reside on the same computer on which the Sterling Control Center engine is located.

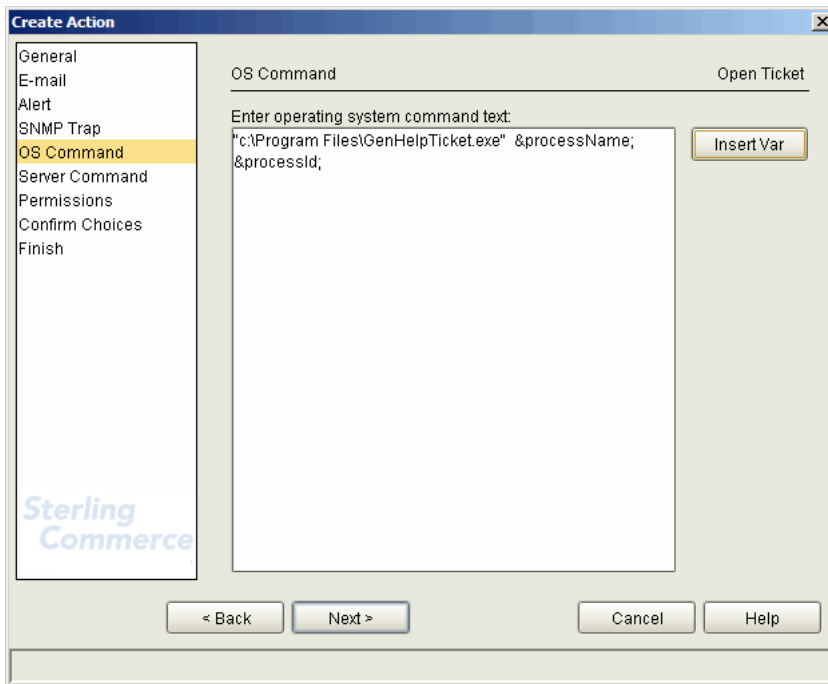
Create an action with a **Name** and **Description** similar to the following.



The screenshot shows the 'Create Action' dialog box with the following details:

- General** (selected in the left sidebar)
- Name:** Open Ticket
- Description:** Call GenHelpTicket to open ticket
- Buttons:** < Back, Next >, Cancel, Help
- Footer:** Description. Up to 255 characters. No special characters. Optional.

Skip to the **OS Command** wizard panel and make the following entry.



Note: The entry must be enclosed in double quotes because of the space in the file path name. Here's an example involving UNIX where no quotation marks are required:

```
cmd /c copy C:\Users\sflan\apkeyf1.txt D:\Users\sflan\apkeyf2.txt
```

Use **Insert Var** to insert the Process Name and Process ID variable names. Click **Next** through the remaining panels and confirm your choices.

Set Up Rules

This chapter discusses how to set up rules to monitor situations frequently seen by Sterling Control Center users.

These examples do not all describe the detailed steps necessary to set up a rule. Some examples describe only the idea behind a solution to a problem. For a detailed explanation of rules and how to set them up, consult the Help system or the *Sterling Control Center System Administration Guide*.

The examples provided are as follows.

General Topics

- How Can I Know When a Process Completes?
- How Can I Know When a Process Has a Problem?
- How Can I Know When a Copy Completes?
- How Can I Know When a Copy Completes Unsuccessfully?
- How Can I Know When a Monitored Server Goes Down?
- How Can I Know When a Monitored Server Starts?
- How Can I Send an E-Mail in the Event of a Critical Database Problem?
- How Can I Generate a Server Down Alert Only After the Server Has Been Down More Than Five Minutes?
- How Can I Generate an Alert When a Managed Server Stops and Handle the Alert Automatically When the Server Restarts?
- How Can I Know When a Connect:Direct Process Goes in the Held in Error Queue?
- How Can I Know When a Managed Server's License is About to Expire?
- How Can I Avoid Receiving Alerts From Both Servers Involved in a Process?
- How Can I Specify More Than One Message ID in the Rule Criteria?
- How Can I Know If a New Version of Objects was Created?
- How Can I Know When Control Center Is Not Monitoring a Server Regularly?

Sterling Integrator

- How Can I Know When a Sterling Integrator Adapter has Stopped?
- How Can I Know When a Sterling Integrator Adapter is Enabled?
- How Can I Know When a Sterling Integrator Adapter Goes Down and Stays Down for More Than Five Minutes?

Sterling File Gateway

- How to Get Notified When a Sterling File Gateway Route Fails?
- How Can a Producer Know When a File Was Delivered to a Sterling Integrator Mailbox?
- How Can a Producer Know if a File Was Delivered to a Consumer's Mailbox?
- How Can a Producer Get Notified if a File Does Not Arrive in Their Mailbox Within 5 Minutes After the Connection is Initiated?

Connect:Enterprise

- How Can I Know When a Connect:Enterprise Daemon Goes Down and Stays Down for More Than Five Minutes?
- How Can I Know When a Connect:Enterprise for UNIX Daemon Stops?
- How Can I Know When a Connect:Enterprise for UNIX Daemon Starts?
- How Can I Know When a Connect:Enterprise for UNIX Offline Add was Performed?
- How Can I Know When a Connect:Enterprise for UNIX AutoConnect Fails?
- How Can I Know When a Connect:Enterprise for UNIX AutoConnect Collect Operation is Performed?
- How Can I Know When a Connect:Enterprise for UNIX AutoConnect Transmit Operation is Performed?
- How Can I Know When a Connect:Enterprise for UNIX Offline Extraction is Performed?
- How Can I Know When a Connect:Enterprise Remote Connection Request Operation was Performed?
- How Can I Know When a Connect:Enterprise Remote Connection Add Operation was Performed?
- How Can I Know When a Connect:Enterprise Remote Connect Connection is Performed?
- How Can I Know When a Connect:Enterprise Remote Connect Delete is Performed?
- How Can I Know When a Connect:Enterprise Remote Connect Directory List is Performed?
- How Can I Know When a Connect:Enterprise Remote Connect Disconnection is Performed?
- How Can I Know When a Connect:Enterprise for z/OS AutoConnect CONNECT Operation is Performed?
- How Can I Know When a Connect:Enterprise for z/OS AutoConnect DISCON Operation is Performed?
- How Can I Know When a Connect:Enterprise for z/OS AutoConnect Send Operation is Performed?
- How Can I Know When a Connect:Enterprise for z/OS AutoConnect Receive Operation is Performed?

How Can I Know When a Process Completes?

For a rule to trigger when a Process completes on a managed server, whether a Sterling Integrator business process, a Connect:Direct Process, or, for example, a batch added by a remote user to a Connect:Enterprise mailbox, set the rule's event type to Process Ended. No other rule attributes need be set; however, to limit the times when this rule would fire, you can set additional values such as an individual server or server group, a Process Name, or a submitter.

The following example illustrates this type of rule setup.

To know when the Process ISITDONE completes on any server in the Server Group known as My Group, specify the following criteria as parameters when you create the rule:

The screenshot shows the 'Create Rule' dialog box with the 'Parameters' tab selected. The 'Process Completed' section contains a table with the following data:

Key	Operator	Value
Event Type	Matches	Process Ended
Servers and Server Groups	Matches	() (MyGroup)
Process Name	Matches	ISITDONE

At the bottom of the dialog, there are buttons for '< Back', 'Next >', 'Cancel', and 'Help'. The Sterling Commerce logo is visible in the bottom left corner. A footer note states: 'Multiple entries allowed. Optional.'

This rule will now fire only when Processes named ISITDONE complete on servers in MyGroup.

How Can I Know When a Process Has a Problem?

For a rule to trigger when a Process completes in error on a managed server, specify an event type of Processes ended, the name of the Process and add another parameter with a key of Return Code, an Operator of Not Equal To, and a Value of 0.

No other rule attributes need be set; however, you can limit the times this rule would fire by setting additional values.

For example, to know when the Process ISITDONE completes in error on any server in the Server Group known as MyGroup, specify the following parameters:

Key	Operator	Value
Event Type	Matches	Process Ended
Process Name	Matches	ISITDONE
Servers and Server Groups	Matches	() (MyGroup)
Return Code	Not equal to	0

This rule will fire only when Processes named ISITDONE complete with a non-zero return code on servers in the server group named MyGroup.

How Can I Know When a Copy Completes?

File transfers are accomplished via what Sterling Control Center interprets as Process steps. For Control Center purposes, this category even includes transfers involving Sterling Integrator servers that don't include a business process. For a rule to trigger when a Process step completes on a managed server (Sterling Integrator Business Process Activity, a transfer handled by a Sterling Integrator adapter such as the CDServerAdapter, a Connect:Direct Process Step, or, for example, a batch added by a remote user to a Connect:Enterprise mailbox), set the rule's event type to Process Step Ended. No other rule attributes need be set; however, to limit the times when this rule will fire, you can set such attributes as server groups, servers, schedules, and parameters like Destination File, Source File, Process Name, or Step Name.

Note: For Connect:Enterprise-managed servers, the Key parameter Submitter equates to the sender's Mailbox ID, while Process Name equates to Batch ID, and Remote Node equates to the recipient's Mailbox ID.

For example, to limit the triggering of your rule to when a file is added to the mailbox named Fred, specify a Key of Remote Node, with an Operator of Matches, and a Value of Fred.

For a Sterling Integrator example: To know when the activity named WaitService in the Sterling Integrator business process ISITDONE completes on the managed server named phoenix-gis 43, specify the following parameters when you create the rule:

Key	Operator	Value
Process Name	Matches	ISITDONE
Step Name	Matches	WaitService
Event Type	Matches	Process Step Ended
Server Id	Matches	phoenix-gis 43

This rule fires only when the Activity named WaitService completes in a business process named ISITDONE on the Sterling Integrator-managed server named phoenix-gis 43.

How Can I Know When a Copy Completes Unsuccessfully?

File transfers are accomplished via what Sterling Control Center interprets as Process Steps. For Control Center purposes, this category even includes transfers involving Sterling Integrator servers that do not include a business process. For a rule to trigger when a Process Step completes in error on a managed server (whether a Sterling Integrator business process activity, a Connect:Direct Process step, or, for example, a batch added by a remote user to a Connect:Enterprise mailbox), you must specify two parameter keys: an event type of Process Step Ended and a Return Code with an Operator of Not Equal To and a Value of 0. No other rule attributes need be set; however, you can further limit the times this rule would fire by setting additional values such as a particular server or a schedule.

For example, to know when the activity called WaitService in business process ISITDONE completes in error on Sterling Integrator-managed server phoenix-gis 43, specify the following parameters when you create the rule:

Key	Operator	Value
Process Name	Matches	ITISDONE
Step Name	Matches	WaitService
Event Type	Matches	Process Step Ended
Server Id	Matches	phoenix-gis 43
Return Code	Not equal to	0

How Can I Know When a Monitored Server Goes Down?

When Sterling Control Center is no longer in communication with a monitored server, it assumes the server has stopped and generates an event of type Server Status with a message ID of CCTR034E.

To be alerted of this condition, create a rule that specifies the following parameter:

Key	Operator	Value
Message ID	Matches	CCTR034E

Specify an action with a severity level that generates an alert (such as severity level 2).

Note: Of course, the Sterling Control Center GUI also relates real-time information about server up or down status. The name of a server in down status appears in red in the list of servers, accompanied by a red down arrow.

How Can I Know When a Monitored Server Starts?

When Sterling Control Center establishes communication with a monitored server, it assumes that the server is started and generates an event of type Server Status with a message ID of CCTR033E.

To be alerted of this condition, create a rule that specifies the following parameter:

Key	Operator	Value
Message ID	Matches	CCTR033E

Note: Of course, the Sterling Control Center GUI also relates real-time information about server up or down status. A server in up status displays in normal text, with an activity graph illustrating any current activity.

How Can I Send an E-Mail in the Event of a Critical Database Problem?

This case involves monitoring for a CJDB026E message, which indicates that an SQL exception has occurred within the Sterling Control Center engine that is not a truncation or duplication exception. When such a problem occurs, Control Center shuts down. In this event, you would probably want to e-mail a responsible party that Control Center has initiated shutdown due to a critical database problem.

To do so, first set up a new action that will send an e-mail to the appropriate party and generate a high-severity alert. Next, create a rule that monitors for a message ID of CJDB026E by specifying the following parameter:

Key	Operator	Value
Message ID	Matches	CJDB026E

Within this rule, associate the e-mail action you just created.

How Can I Generate a Server Down Alert Only After the Server Has Been Down More Than Five Minutes?

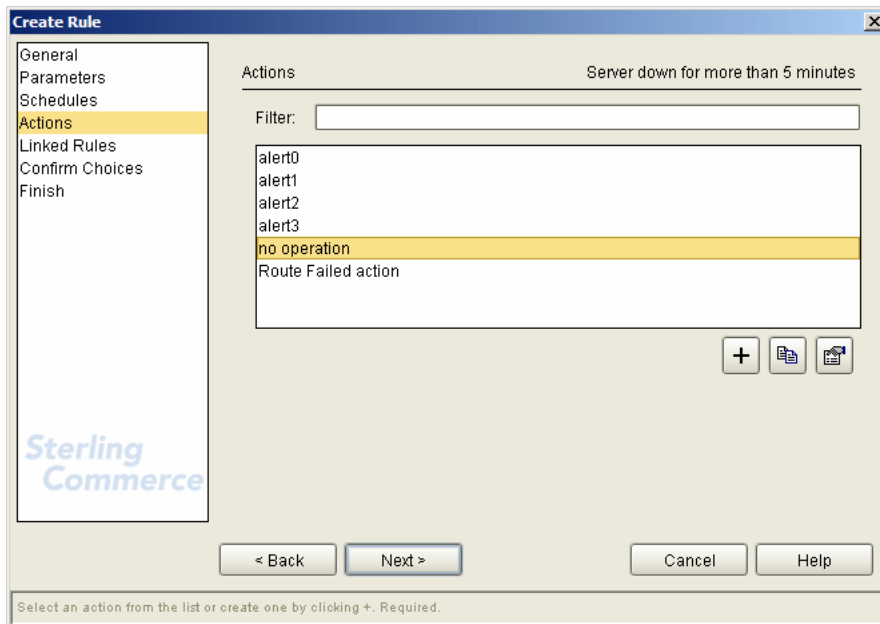
When Sterling Control Center can no longer communicate with a monitored server, it assumes the server has stopped, and generates an event of type Server Status with a message ID of CCTR034E. Also, when Sterling Control Center establishes communication with a monitored server, it assumes the server is started and generates an event of type Server Status with a message ID of CCTR033E.

This information can be used in conjunction with a linked rule to generate an alert for a server down condition only if the server has been down for more than, say, five minutes, thus giving a system administrator a chance to restart the server before any alert is generated. A rule like this would be useful when brief server outages are of minor consequence and the only time you would like an alert to be generated is when a longer-term outage occurs.

Note that a linked rule is nothing more than a normal rule with additional attributes.

To construct a rule to let you know when a server is down more than five minutes, first build a rule that monitors for the message that Sterling Control Center generates when it thinks a server has stopped by specifying a key of Message ID, an Operator of Matches, and a Value of CCTR034E. You can specify additional parameters to limit when this rule is triggered.

Select the No Operation action, since you don't want to generate an alert when the server is first discovered to be down, only after it has been down for more than five minutes.



Select Enabled on the Linked Rules wizard panel that appears next.

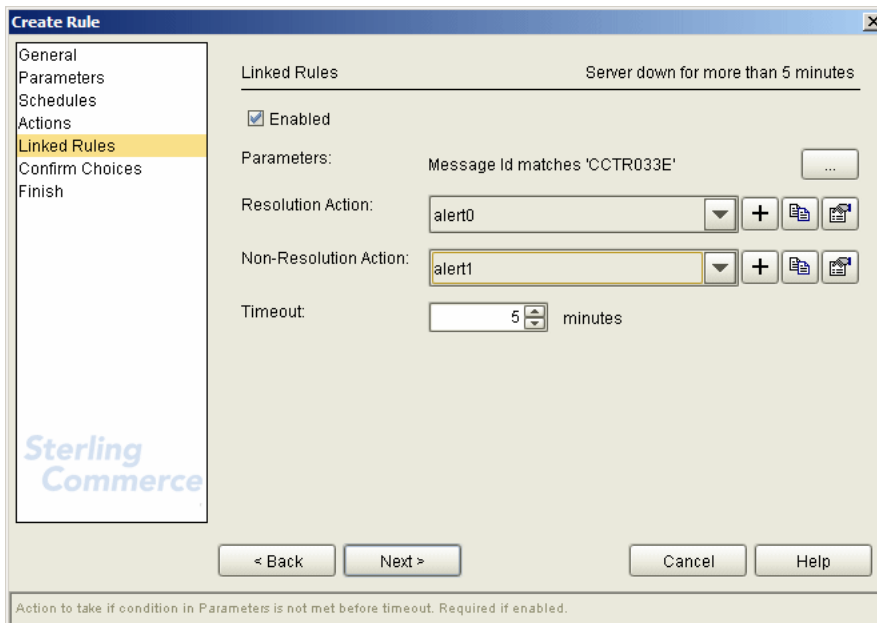
Specify a following parameter key of Message ID, an operator of Matches, and a value of CCTR033E, as an event that will be generated by the Sterling Control Center engine when it finds that a managed server has started. You can also specify other parameters to refine the criteria under which CCTR033E events are matched by the linked part of the rule.

How Can I Generate a Server Down Alert Only After the Server Has Been Down More Than Five Minutes?

Since you selected No Operation for the initial action, choose the same for the Resolution action—nothing needs to be done if the server comes back up within five minutes—and an appropriate action, such as alert1, for the Non-Resolution action.

Note: If you selected an alert action for the initial action, you should choose alert0 for the resolution action, which tells Sterling Control Center to clear the first alert, and again an appropriate action for the Non-Resolution action, such as alert1. Doing this generates a lower-severity alert when the server is first discovered to be down and a higher-severity alert if the server remains down for more than the allowed amount of time.

Set the Timeout value to the time the Control Center engine should wait before taking either the Resolution or the Non-Resolution action, in this case five minutes.



Create Rule

General
Parameters
Schedules
Actions
Linked Rules
Confirm Choices
Finish

Linked Rules: Server down for more than 5 minutes

Enabled

Parameters: Message Id matches 'CCTR033E'

Resolution Action: alert0

Non-Resolution Action: alert1

Timeout: 5 minutes

< Back Next > Cancel Help

Action to take if condition in Parameters is not met before timeout. Required if enabled.

How Can I Generate an Alert When a Managed Server Stops and Handle the Alert Automatically When the Server Restarts?

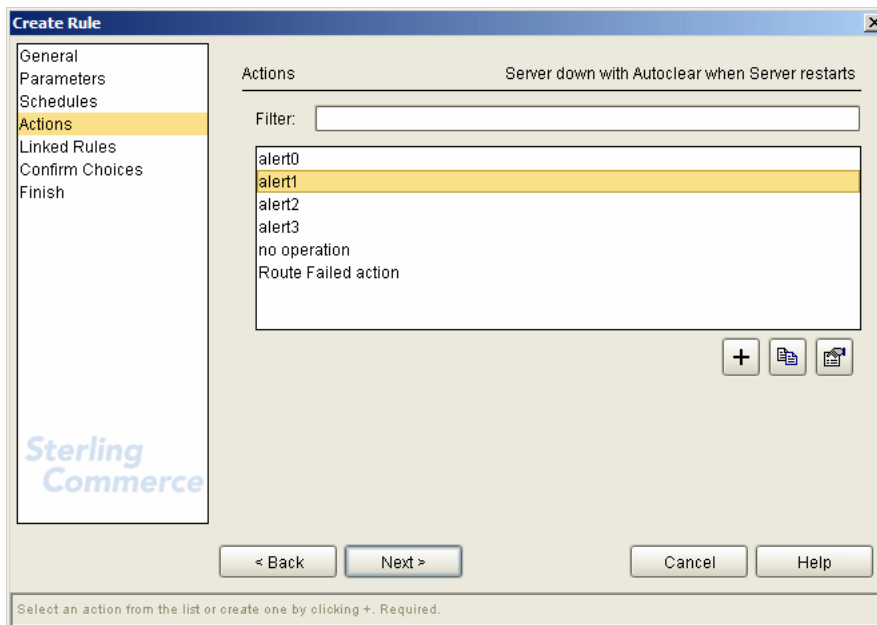
When Sterling Control Center can no longer communicate with a monitored server, it assumes the server has stopped and generates an event of type Server Status with a message ID of CCTR034E. Also, when Sterling Control Center establishes communication with a monitored server, it assumes the server is started and generates an event of type Server Status with a message ID of CCTR033E.

This information can be used in conjunction with a linked rule to generate an alert for a server down condition and to automatically clear any alert generated if and when the server is restarted.

Note that a linked rule is nothing more than a normal rule with additional attributes.

To construct a rule to let you know when a server is down, first build a rule with a parameter using a Key of Message ID, an Operator of Matches, and a Value of CCTR034E—the event generated by the Sterling Control Center engine when it thinks a server has stopped. You can specify additional parameters to limit when this rule is triggered.

Select alert1 as the Action to generate an alert of severity 1 when the Server stops.



Select Enabled on the Linked Rules wizard panel that appears next.

For linked rule parameters, specify a parameter with a key of Message ID, an Operator of Matches, and a Value of CCTR033E, as an event that will be generated by the Sterling Control Center engine when it finds that a managed server has started. Other parameters may also be specified as necessary to ensure only the appropriate CCTR033E events are matched.

Since you want to clear the alert generated when the server down occurred when the system sees that the server has restarted, you should choose alert0 for the Resolution action, and the action No Operation for the Non-Resolution action.

How Can I Generate an Alert When a Managed Server Stops and Handle the Alert Automatically When the Server Restarts?

Select alert0 for the Non-Resolution action as well if you always want the alert cleared. The Non-Resolution action will not occur until the number of minutes specified for Timeout has transpired.

Set the Timeout value to the time the Control Center engine should wait before taking either the Resolution or the Non-Resolution action. That is, set the Timeout value to the maximum time you feel it will take for a server outage to be handled.

Create Rule

General
Parameters
Schedules
Actions
Linked Rules
Confirm Choices
Finish

Linked Rules: Server down with Autoclear when Server restarts

Enabled

Parameters: Message Id matches 'CCTR033E'

Resolution Action: alert0

Non-Resolution Action: alert0

Timeout: 1440 minutes

Sterling Commerce

< Back Next > Cancel Help

Timeout in min. Default 5. Minimum 1. Maximum 64,546.

How Can I Know When a Connect:Direct Process Goes in the Held in Error Queue?

For a rule to trigger when a Connect:Direct Process has been placed in the Hold Queue after it received an error (status = HE, Held in Error), specify a parameter with a Key of Message Id, with an Operator of Matches, and a Value of CNCD061I.

Process moved to Hold Queue

Key	Operator	Value
Message Id	Matches	CNCD061I

Multiple entries allowed. Optional.

Values for Servers and Server Groups and additional parameters may be specified to make this rule more specific if desired.

When this rule is associated with an action that issues the Connect:Direct Release Process command, whenever Sterling Control Center finds a Connect:Direct Process has moved to the Hold Queue with an HE status, it tells the managed server to release it to run again.

How Can I Know When a Connect:Enterprise for UNIX Daemon Stops?

When Sterling Control Center finds that a master type Daemon on a Connect:Enterprise for UNIX-monitored server has transitioned from running to not running, that is, the daemon has stopped—it generates an event of type Server Status with a message ID of CNCE115I.

You can simply write a rule that specifies a parameter with a Key of Message ID along with a Value of CNCE115I, or, to limit when the rule is triggered, you can also specify additional parameters, as shown below.

Multiple entries allowed. Optional.

How Can I Know When a Connect:Enterprise for UNIX Daemon Starts?

When Sterling Control Center finds that a protocol or other type of daemon on a Connect:Enterprise for UNIX-monitored server has transitioned from being down to running, it generates an event of type Server Status with a message ID of CNCE114I.

How Can I Know When a Connect:Enterprise Daemon Goes Down and Stays Down for More Than Five Minutes?

When a daemon goes down, a Server Status event with a message ID of CNCE115I is generated. When a daemon starts, a Server Status event with a message ID of CNCE114I is generated.

Using this knowledge a linked rule can be built to accomplish the objective of alerting you only if the daemon goes down and stays down for more than five minutes.

The first part of the rule is simply to watch for the message ID of CNCE115I. The action taken could be No Operation, or a low-severity alert action.

Select Enabled on the Linked Rules dialog that appears next.

Specify a Message ID value of CNCE114I as a Linked Rules parameter.

If you selected No Operation for the initial action, choose the same for the Resolution action, and an appropriate action, such as alert1, for the Non-Resolution action.

If you selected an alert action, such as alert2, for the initial action, you should choose alert0 for the Resolution action, which tells Sterling Control Center to clear the first alert generated, and an appropriate action for the Non-Resolution action.

Set the Timeout value to the time the Control Center engine should wait before taking either the Resolution or the Non-Resolution action.

The screenshot shows the 'Create Rule' dialog box with the following configuration:

- General:** Parameters, Schedules, Actions, **Linked Rules**, Confirm Choices, Finish
- Linked Rules:** Daemon down more than 5 minutes
- Enabled
- Parameters: Message Id matches 'CNCE114I'
- Resolution Action: no operation
- Non-Resolution Action: alert1
- Timeout: 5 minutes

Buttons: < Back, Next >, Cancel, Help

Footer: Action to take if condition in Parameters is not met before timeout. Required if enabled.

How Can I Know When a Sterling Integrator Adapter has Stopped?

When Sterling Control Center finds that a Sterling Integrator Adapter on a Sterling Integrator-monitored server has transitioned from being up to being down, an event of type Server Status with a message ID of CGIS011I is generated.

You can write a rule that specifies a parameter with a Key of Message ID, an Operator of Matches, and a Value of CGIS011I. To limit the circumstances under which the rule is triggered, you can also specify values for Server Groups, Servers, as well as other parameters, as shown below.

Multiple entries allowed. Optional.

How Can I Know When a Sterling Integrator Adapter is Enabled?

When Sterling Control Center finds that a Sterling Integrator Adapter on a Sterling Integrator-monitored server has transitioned from being down to up, that is, enabled, an event of type Server Status with a message ID of CGIS010I is generated.

How Can I Know When a Sterling Integrator Adapter Goes Down and Stays Down for More Than Five Minutes?

When a Sterling Integrator adapter goes from enabled to disabled on a Sterling Integrator-managed server, a Server Status event with a message ID of CGIS011I is generated, and when a Sterling Integrator adapter starts up/is enabled, a Server Status event with a message ID of CGIS010I is generated.

Using this knowledge, you can build a linked rule to accomplish the objective of alerting you only if the adapter goes down and stays down for more than five minutes.

The first part of the rule is to watch for the message ID of CGIS011I. The action taken could be No Operation, or a low severity alert action.

Select Enabled on the Linked Rules dialog that appears next.

As a Linked Rules parameter, specify a Message ID value of CGIS010I to match on the event when a Sterling Integrator adapter starts up.

If you selected No Operation for the initial action, choose No Operation for the Resolution action also, and an appropriate action, such as alert1, for the Non-Resolution action.

If you selected an alert action, such as alert2, for the initial action, you should choose alert0 for the resolution action. This tells Sterling Control Center to clear the first alert generated, and an appropriate action for the Non-Resolution action.

Set the Timeout value to the time the Control Center engine should wait before taking either the Resolution or the Non-Resolution action.

Create Rule

General
Parameters
Schedules
Actions
Linked Rules
Confirm Choices
Finish

Linked Rules: SI Adapter down more than 5 minutes

Enabled

Parameters: Message Id matches 'CGIS010I'

Resolution Action: no operation

Non-Resolution Action: alert1

Timeout: 5 minutes

Sterling Commerce

< Back Next > Cancel Help

Action to take if condition in Parameters is not met before timeout. Required if enabled.

How Can I Know When a Connect:Enterprise Remote Connection Request Operation was Performed?

Look for a Process End or Process Step End event, with a message ID of CNCE005I.

Caution: If you specify only a message ID as a parameter, your rule will be triggered twice per Remote Connection Request operation.

How Can I Know When a Connect:Enterprise Remote Connection Add Operation was Performed?

Look for a Process End or Process Step End event, with a message ID of CNCE003I.

Caution: If you specify only a message ID as a parameter, your rule will be triggered twice per Remote Connection Request operation.

How Can I Know When a Connect:Enterprise for UNIX Offline Add was Performed?

Look for a Process End or Process Step End event, with a message ID of CNCE012I.

Note: This currently works only for Connect:Enterprise for UNIX nodes.

Caution: If you specify only a message ID as a parameter, your rule will be triggered twice per Remote Connection Request operation.

How Can I Know When a Connect:Enterprise for UNIX AutoConnect Fails?

Look for a message ID of CNCE113I.

How Can I Know When a Connect:Enterprise Remote Connect Connection is Performed?

Look for a message ID of CNCE009I.

How Can I Know When a Connect:Enterprise Remote Connect Connection is Performed?

Look for a message ID of CNCE009I.

How Can I Know When a Connect:Enterprise Remote Connect Delete is Performed?

Look for a message ID of CNCE007I.

How Can I Know When a Connect:Enterprise Remote Connect Directory List is Performed?

Look for a message ID of CNCE006I.

How Can I Know When a Connect:Enterprise Remote Connect Disconnection is Performed?

Look for a message ID of CNCE010I.

How Can I Know When a Connect:Enterprise for UNIX Offline Extraction is Performed?

Look for a Process Ended or Process Step Ended event, with a message ID of CNCE011I.

Note: This currently works only for Connect:Enterprise for UNIX nodes.

Caution: If you specify only a message ID as a parameter, your rule will be triggered twice per Remote Connection Request operation.

How Can I Know When a Connect:Enterprise for z/OS AutoConnect CONNECT Operation is Performed?

Look for a message ID of CNCE209I.

Note: This currently works only for Connect:Enterprise for z/OS nodes.

How Can I Know When a Connect:Enterprise for z/OS AutoConnect DISCON Operation is Performed?

Look for a message ID of CNCE210I.

Note: This currently works only for Connect:Enterprise for z/OS nodes.

How Can I Know When a Connect:Enterprise for z/OS AutoConnect Send Operation is Performed?

Look for a Process Ended or Process Step Ended event, with a message ID of CNCE001I.

Note: This currently works only for Connect:Enterprise for z/OS nodes.

Caution: If you specify only a message ID as a parameter, your rule will be triggered twice per Remote Connection Request operation.

How Can I Know When a Connect:Enterprise for z/OS AutoConnect Receive Operation is Performed?

Look for a Process Ended or Process Step Ended event, with a message ID of CNCE002I.

Note: This currently works only for Connect:Enterprise for z/OS nodes.

Caution: If you specify only a message ID as a parameter, your rule will be triggered twice per Remote Connection Request operation.

How Can I Know When a Connect:Enterprise for UNIX AutoConnect Collect Operation is Performed?

Look for a Process Ended or Process Step Ended event, with a message ID of CNCE013I.

Note: This currently works only for Connect:Enterprise for UNIX nodes.

Caution: If you specify only a message ID as a parameter, your rule will be triggered twice per Remote Connection Request operation.

How Can I Know When a Connect:Enterprise for UNIX AutoConnect Transmit Operation is Performed?

Look for a Process Ended or Process Step Ended event, with a message ID of CNCE014I.

Note: This currently works only for Connect:Enterprise for UNIX nodes.

Caution: If you specify only a message ID as a parameter, your rule will be triggered twice per Remote Connection Request operation.

How Can I Know When a Managed Server's License is About to Expire?

When a managed server's license is about to expire, an event of type Server License with a message ID of CCNS004E is generated every day at midnight (and whenever Sterling Control Center establishes a new connection with the server).

How Can I Avoid Receiving Alerts From Both Servers Involved in a Process?

If both servers involved in a failed Process are managed by Control Center, two alerts may be generated. The redundant alert can be eliminated by setting up a rule that causes Control Center to ignore one of the two alert conditions.

Create a rule that specifies the following parameters:

Key	Operator	Value
Event Type	Matches	Process Step Ended
Return Code	Not equal to	0
Local Node (P or S)	Matches	S

Specify an action of “No operation” and finish by confirming your choices.

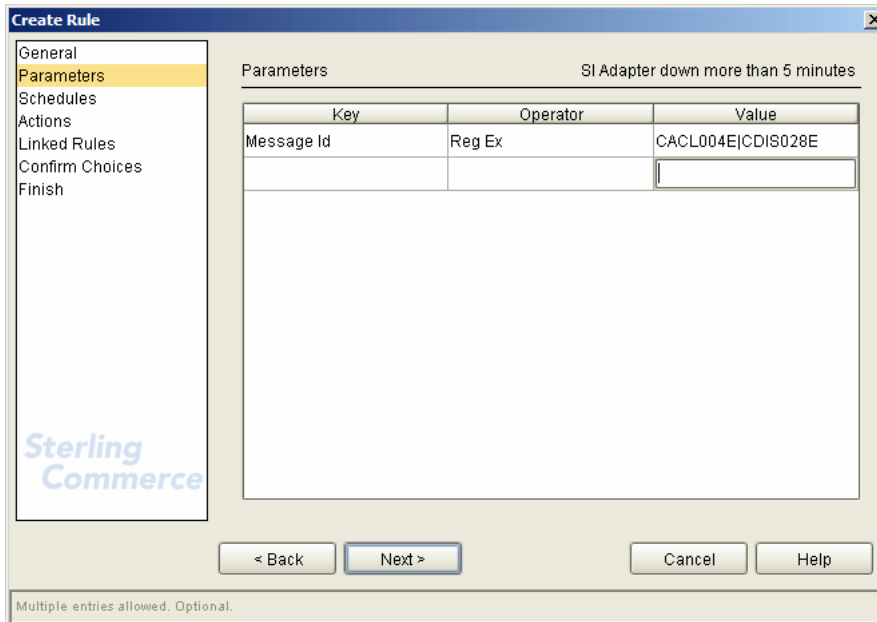
Note: Except for the additional parameter of “Local Node (P or S) Matches S,” this rule is essentially a duplicate of the Bad Return Code rule that ships with Control Center. To work, it must be placed above the Bad Return Code rule in the Rules listing; i.e., it must have a higher priority in the listing.

Caution: This technique works except in the following situations:

- If a PNODE = SNODE Process step fails, two alerts are still generated.
 - No alerts are generated if the PNODE server involved in the Process is unmanaged. Circumvent this shortcoming by preceding the rule in the Rules listing with one that specifies a particular server as the SNODE for a Process.
-

How Can I Specify More Than One Message ID in the Rule Criteria?

Use regular expressions to specify more than one Message ID in the rule, as in this illustration:



Key	Operator	Value
Message Id	Reg Ex	CACL004E CDIS028E

Parameters: SI Adapter down more than 5 minutes

Multiple entries allowed. Optional.

For details, see the *Regular Expressions* appendix of the *System Administration Guide*.

How Can I Know If a New Version of Objects was Created?

When a new version of objects is created for a server in the Control Center object repository a Server command event is generated with one of the following message IDs:

- CCFG191I – New version of Netmap Nodes created
- CCFG192I – New version of Netmap Modes created
- CCFG193I – New version of Netmap Communication Paths created
- CCFG194I – New version of Init Params created
- CCFG195I – New version of Translation tables created
- CCFG196I – New version of User Proxies created
- CCFG197I – New version of User Auth entries created
- CCFG198I – New version of Secure+ Nodes created
- CCFG199I – New version of Secure+ Key Certificates created
- CCFG200I – New version of Secure+ Trusted Certificates created

You may write multiple rules, one for each message ID, or write one rule that specifies one, or more, of these message IDs using RegEx as the Operator. See *How Can I Specify More Than One Message ID in the Rule Criteria?* on page 75.

How Can I Know When Control Center Is Not Monitoring a Server Regularly?

To know when monitoring of a server is not taking place on a regular basis, use the new built-in rule, *Monitoring rate out of compliance*. The monitor rest time is the amount of time (usually in seconds) Control Center waits before polling the server to check status and collect statistics. The monitor rest time is used to determine whether monitoring is in or out of compliance.

Note: To be on the safe side, make a copy of the built-in rule, and then edit and enable your copy, instead of modifying and using the built-in rule.

How to Get Notified When a Sterling File Gateway Route Fails?

Set up a rule to generate an alert, and in the rule specify the following parameter:

Key	Operator	Value
Message Id	Matches	FG_0456

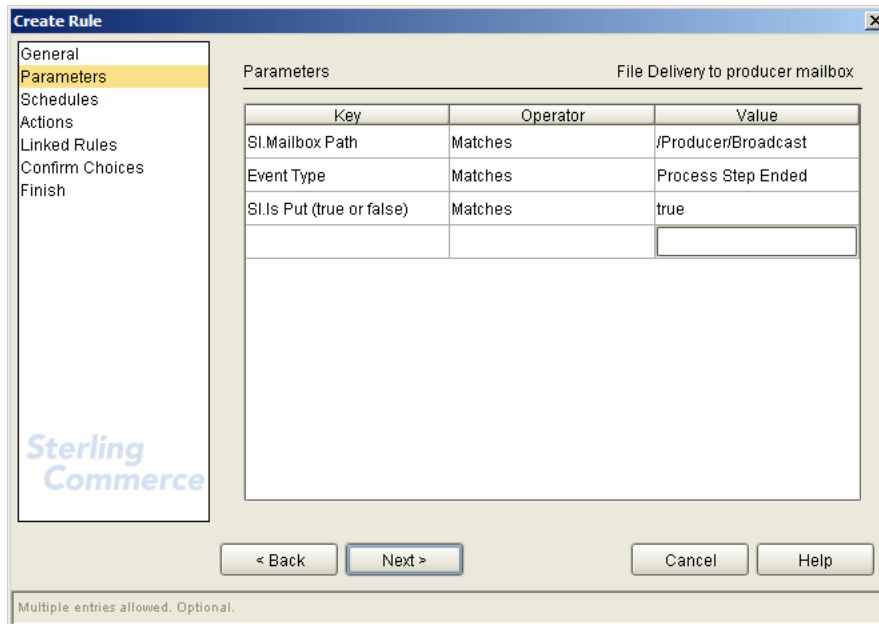
Route failed

Multiple entries allowed. Optional.

Sterling File Gateway “event codes” are mapped to Control Center event message IDs. To look up a particular event code, go to the Sterling File Gateway documentation. Event Code FG_0456 is named RouteFailed and is the appropriate value to use for this scenario.

How Can a Producer Know When a File Was Delivered to a Sterling Integrator Mailbox?

Assuming files are delivered using one of the monitored protocols, set up a rule with the following parameters substituting an appropriate value for your mailbox location:



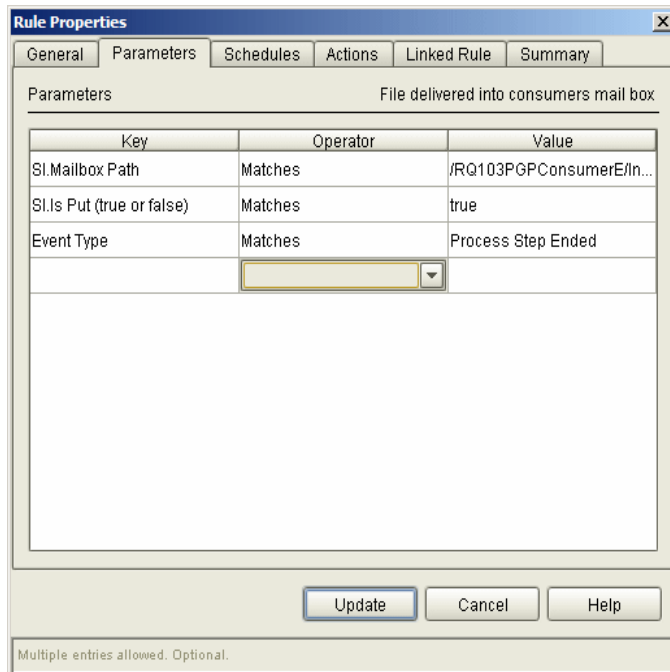
The screenshot shows the 'Create Rule' dialog box with the 'Parameters' tab selected. The rule is titled 'File Delivery to producer mailbox'. The parameters are defined in a table:

Key	Operator	Value
SI.Mailbox Path	Matches	/Producer/Broadcast
Event Type	Matches	Process Step Ended
SI.Is Put (true or false)	Matches	true

At the bottom of the dialog, there are buttons for '< Back', 'Next >', 'Cancel', and 'Help'. The Sterling Commerce logo is visible in the bottom left corner of the dialog. A note at the bottom states: 'Multiple entries allowed. Optional.'

How Can a Producer Know if a File Was Delivered to a Consumer's Mailbox?

Set up a rule with the following parameters substituting an appropriate value for your mailbox location:



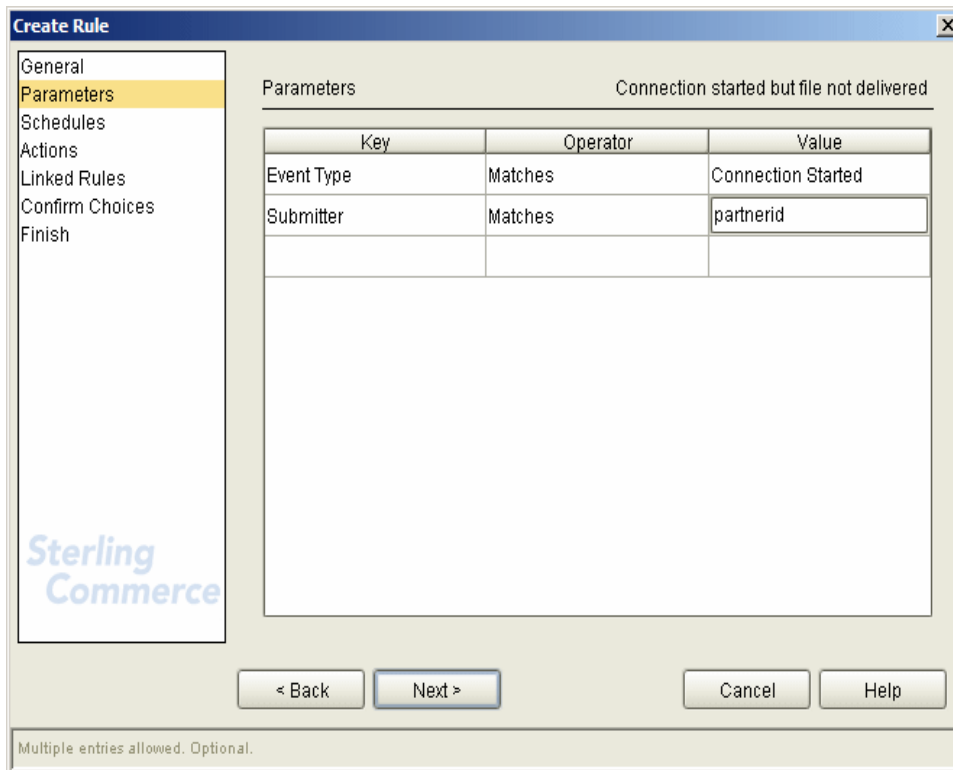
Key	Operator	Value
SI.Mailbox Path	Matches	/RQ103PGPConsumerE/In...
SI.Is Put (true or false)	Matches	true
Event Type	Matches	Process Step Ended

Update Cancel Help

Multiple entries allowed. Optional.

How Can a Producer Get Notified if a File Does Not Arrive in Their Mailbox Within 5 Minutes After the Connection is Initiated?

To implement this scenario, set up a linked rule and specify the following parameters substituting an appropriate value for the submitter in your situation:

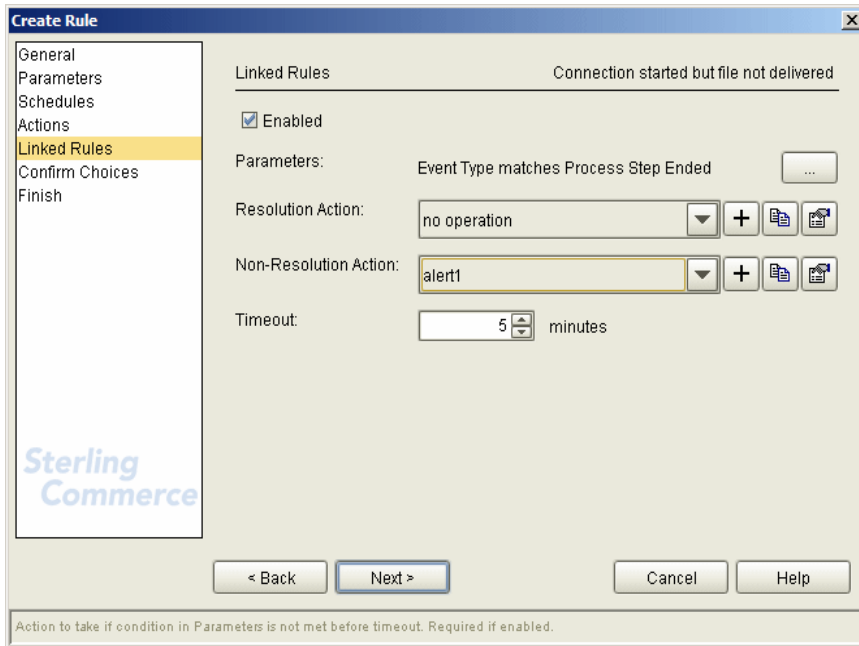


Key	Operator	Value
Event Type	Matches	Connection Started
Submitter	Matches	partnerid

On the Actions screen, select no operation.

Next, define the following parameters on the Linked Rules screen:

- Enabled—Check to enable.
- Parameters—Event Type Matches Process Step Ended
- Resolution action—no operation
- Non-Resolution action—alert1
- Timeout—5 minutes to define the period of time to wait before taking the non-resolution action (alert1 which indicates a high alert)



Troubleshooting Rules

Troubleshoot Rules

Rules can fail to work as expected for many reasons. Here we define *working* as being triggered by the conditions under which a rule is expected to be triggered. Here are some examples of why a rule might not be triggered as expected:

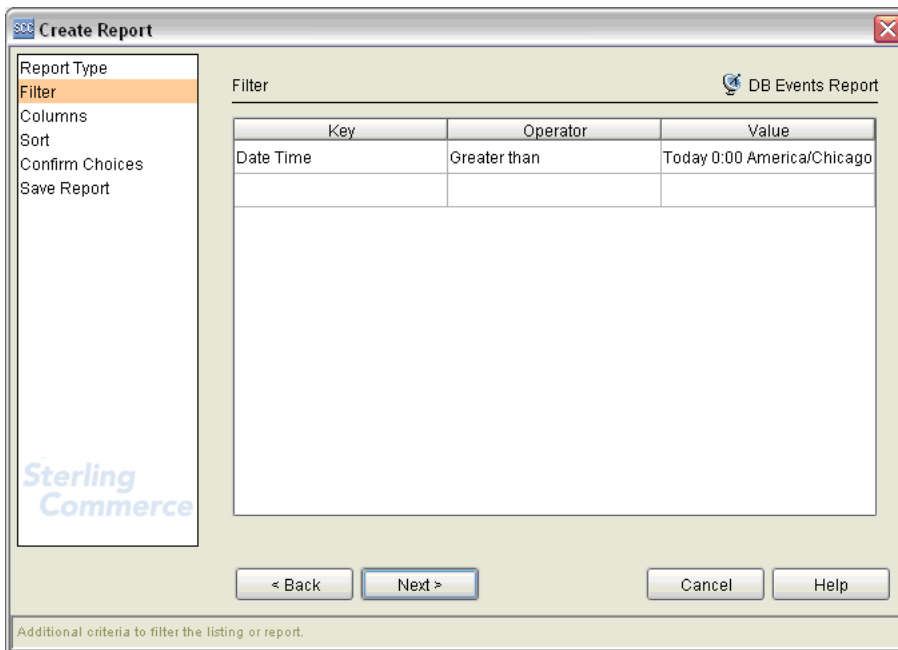
- A rule will not be triggered if the wrong Event Type is selected even though all of the other parameters are correct.
- A rule will not be triggered if no events matching its parameters have been generated.
- A rule will not be triggered even if an event matches its parameters if the event occurred outside of the schedule associated with the rule.
- A rule will not be triggered if the matching event has already triggered a rule that appears higher in the list.
- A rule must be enabled in order to be triggered.
- Back-level versions of managed servers, that is, managed servers not on current maintenance, may not generate statistics records, or statistics records with the needed information, for the events constructed by Sterling Control Center to have the information needed to trigger the rule.
- Process events (Process Started, Process Step Started, Process Step Ended, and Process Ended) are all generated at the same time by Control Center for Connect:Enterprise for z/OS, Connect:Enterprise for UNIX, and most native FTP servers because Control Center only “sees” the file transfer ending event. Be aware of this when writing linked rules for process events for these types of servers.

One method you can use to determine the parameter that should be defined in a rule is to cause the condition that you want to trigger the rule on a managed server. After Sterling Control Center has captured the data generated by the condition, and written an event to its events database table, use the Control Center Database Events report to capture the event’s exact contents in order to examine them.

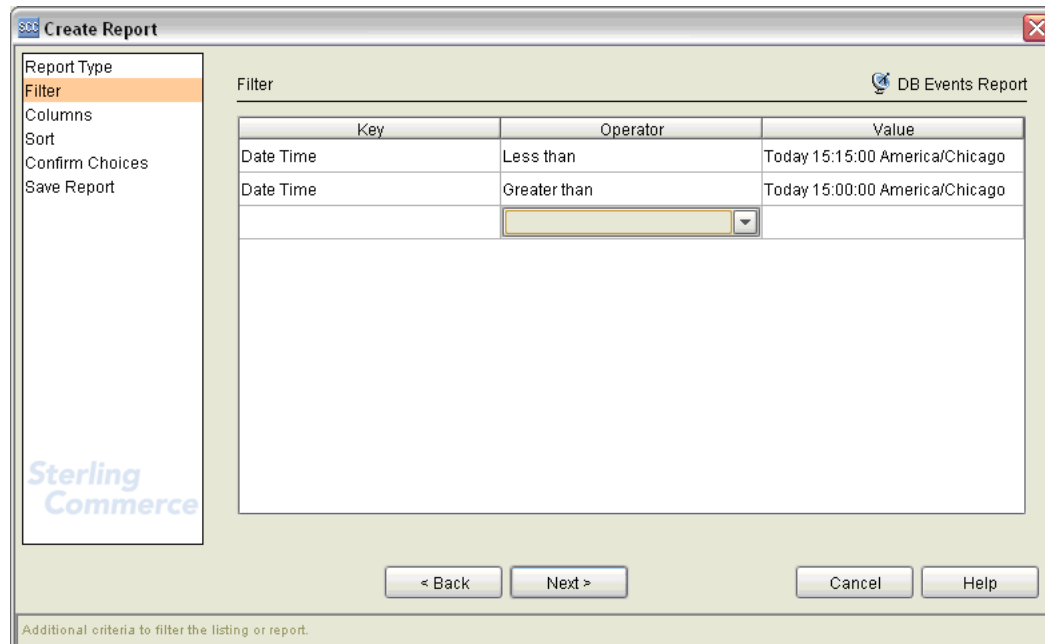
First, define a new report. In the create report wizard, after selecting a Report Type of **DB Events Report**, click **Next**.

For the DB Events Report Filter parameters, specify as much information as you can to limit the amount of data being returned.

At a minimum, specify a starting and an ending Date/Time for the report:

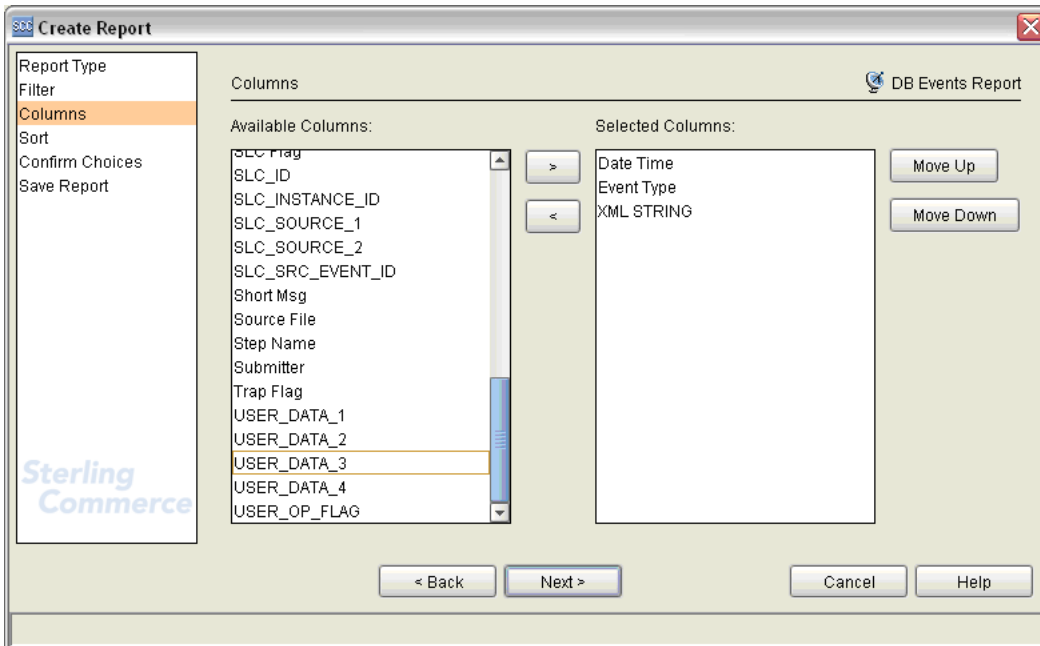


The criteria below cause all events generated between 15:00 and 15:15 to be retrieved. You may specify additional filter keys as you see fit to limit the size of the report.

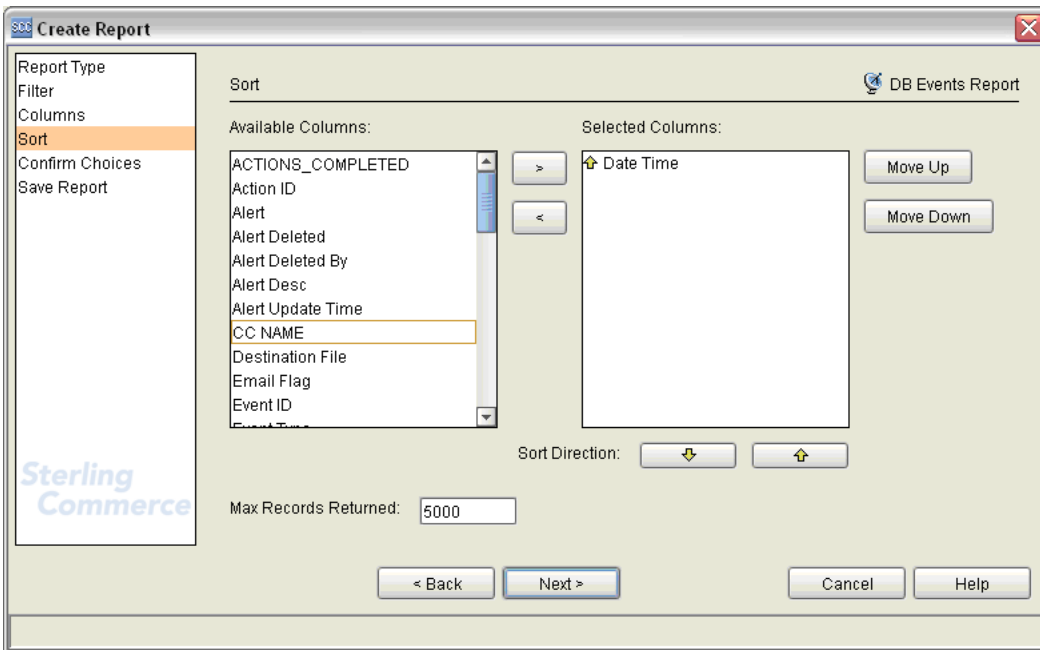


Click **Next** when you've finished specifying filter criteria.

The Available Columns wizard panel is displayed next.



Select Date Time, Event Type, and XML_STRING columns and click **Next**.



The Sort wizard panel is displayed. Select Date Time and click **Next**, and then run the report.

The report generated contains the events generated during the specified time frame that also match any additional filter criteria specified. The XML_STRING value shows the contents of the actual Control Center event. From these values you should be able to determine the parameters necessary for your rule to be triggered as expected.

For example, say you wanted to write a rule to be triggered when a Connect:Direct server stops.

First, while Control Center is monitoring a server, stop the server.

Second, create a Database Events Report over the time frame the server was stopped.

You may get output similar to the following report:

The screenshot shows a window titled "Sterling Control Center Reports Viewer" displaying an "Events Report". The report is a table with three columns: "Date Time", "Event Type", and "XML STRING". The XML STRING column contains detailed event data, including process IDs, node names, and message IDs. The report is paginated, showing "Page 1 of 2" at the bottom.

Date Time	Event Type	XML STRING
2006/09/01 15:11:12.000	12	<event><processId>0</processId><nodeId>dander2k41</nodeId><origNode>DANDER2K41</origNode><action sFlag>115714149599</actionsFlag><nodeType>1</nodeType><messageId>LSM0421</messageId><seqNum>75834126579337963</seqNum><returnCode>0</returnCode><eventId>75834126579337963</eventId><date Time>2006/09/01 20:11:12.000</dateTime><recordId>NUTR</recordId><nodeName>DANDER2K41</nodeName><eventType>12</eventType ><shortText>ConnectDirect F shut-down begun.</shortText></event>
2006/09/01 15:11:12.000	12	<event><processId>0</processId><nodeId>dander2k41</nodeId><origNode>DANDER2K41</origNode><action sFlag>1157141496015</actionsFlag><nodeType>1</nodeType><messageId>LSM0421</messageId><seqNum>75834126579337964</seqNum><returnCode>0</returnCode><eventId>75834126579337964</eventId><date Time>2006/09/01 20:11:12.000</dateTime><recordId>NUTR</recordId><nodeName>DANDER2K41</nodeName><eventType>12</eventType ><shortText>ConnectDirect F shut-down begun.</shortText></event>
2006/09/01 15:11:14.653	5	<event><ruleId>Server down with Autoclear when Server restarts</ruleId><nodeId>dander2k41</nodeId><actionsFlag>1157141474653</actionsFlag><ruleInstancId>75 834126579337953</ruleInstancId><nodeType>1</nodeType><messageId>CCTR034E</messageId><seqNum>75834126579337953</seqNum><eventId>75834126579337953</eventId><dateTime>2006/09/01 20:11: 14.653</dateTime><nodeName>DANDER2K41</nodeName><alert>1</alert><actionId>[alert1] </actionId><trapFlag>1157141474653</trapFlag><eventType>5</eventType><shortText>CCTR034E Server is down.</shortText></event>
2006/09/01 15:11:35.000	5	<event><dateTime>2006/09/01 20:11:35.000</dateTime><eventId>75834126579337966</eventId><nodeId>dander2k41</nodeId><actionsFlag>1157141496 015</actionsFlag><nodeName>DANDER2K41</nodeName><nodeType>1</nodeType><messageId>CNGD051 I</messageId><eventType>5</eventType><seqNum>75834126579337966</seqNum><shortText>CNGD051 License message: Connect:Direct Initialization using Emergency Key.</shortText></event>
2006/09/01 15:11:35.405	5	<event><dateTime>2006/09/01 20:11:35.405</dateTime><eventId>75834126579337955</eventId><nodeId>dander2k41</nodeId><actionsFlag>1157141495 577</actionsFlag><nodeName>DANDER2K41</nodeName><nodeType>1</nodeType><messageId>CCNS005 E</messageId><eventType>5</eventType><seqNum>75834126579337955</seqNum><shortText>CCNS005E License obtained.</shortText></event>

Page 1 of 2

Fri Sep 01 15:16:39 CDT

Page 1 of 2

Locate the event generated by the server down condition in the report.

In this case the event looked like this:

```
<event><dateTime>2006/09/01
20:19:46.392</dateTime><eventId>75834126579338262</eventId><nodeId>dander2k41</nodeId><actionsFlag
>1157141986439</actionsFlag><nodeName>DANDER2K41</nodeName><nodeType>1</nodeType><messag
eId>CCTR034E</messageId><eventType>5</eventType><seqNum>75834126579338262</seqNum><shortTex
t>CCTR034E Server is down.</shortText></event>
```

From this event, you can see that when a server goes down, the message ID is CCTR034E.

You can use this information to write a server down rule. In creating the server down rule, specify a parameter with a Key of Message ID, an Operator of Matches, and Value of CCTR034E.

Set Up Schedules

Because schedules are often used in conjunction with service-level criteria (SLCs), many examples in *Set Up SLCs* on page 93 detail the setting up of a schedule for a specific purpose. This chapter shows how to set up a schedule to deal with a problem common to all rules, that of triggering during maintenance periods when servers are expected to be down.

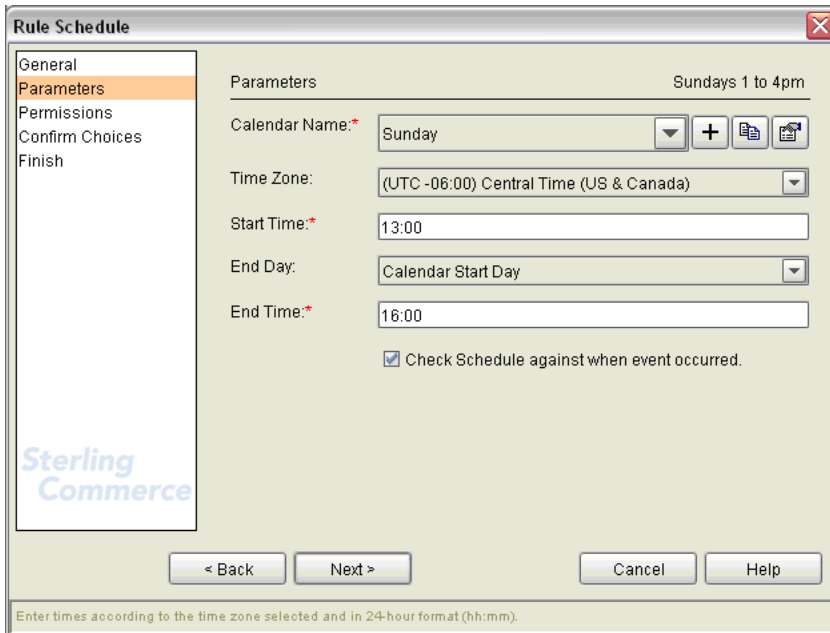
For a detailed explanation of schedules and how to set them up, consult the Help system or the *Sterling Control Center System Administration Guide*.

How Can I Build a Schedule To Prevent Rules From Triggering During My Maintenance Windows?

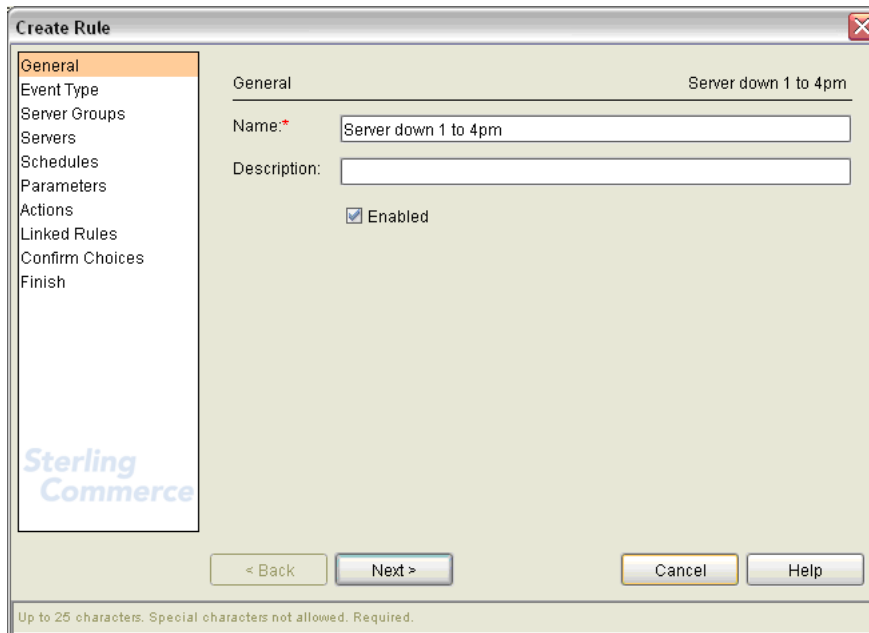
Let's say you have weekly maintenance windows on Sundays from 1pm to 4pm and you always want to know when servers go down *except* during these maintenance windows.

First, create a Server Down rule, as described in *How Can I Know When a Monitored Server Goes Down?* on page 43, with no associated schedule.

Second, create a rule schedule that specifies all Sundays from 1 to 4pm. This is easy to do, because a calendar by the name of Sunday ships with Sterling Control Center which specifies all Sundays in a year.

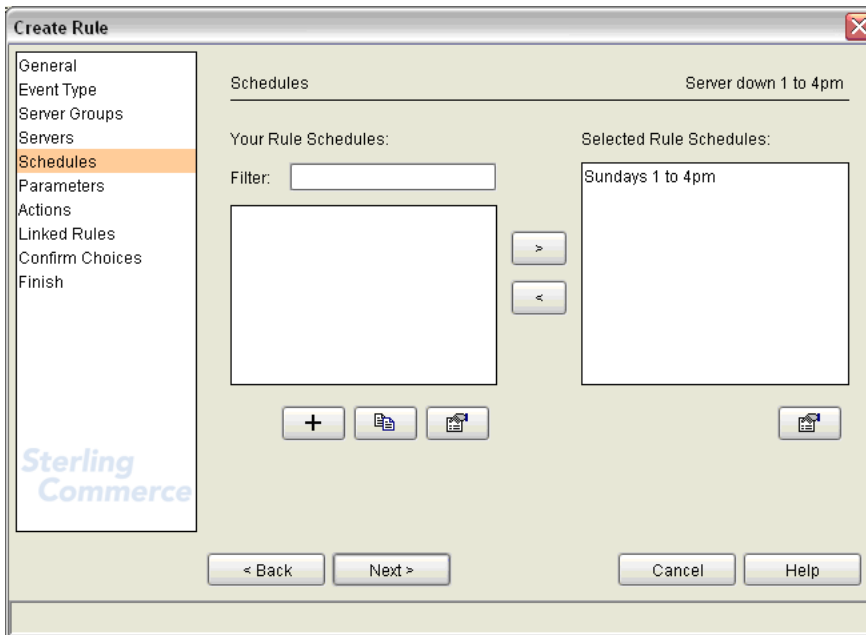


Third, duplicate the server down rule you created before and name it Server Down 1 to 4pm.

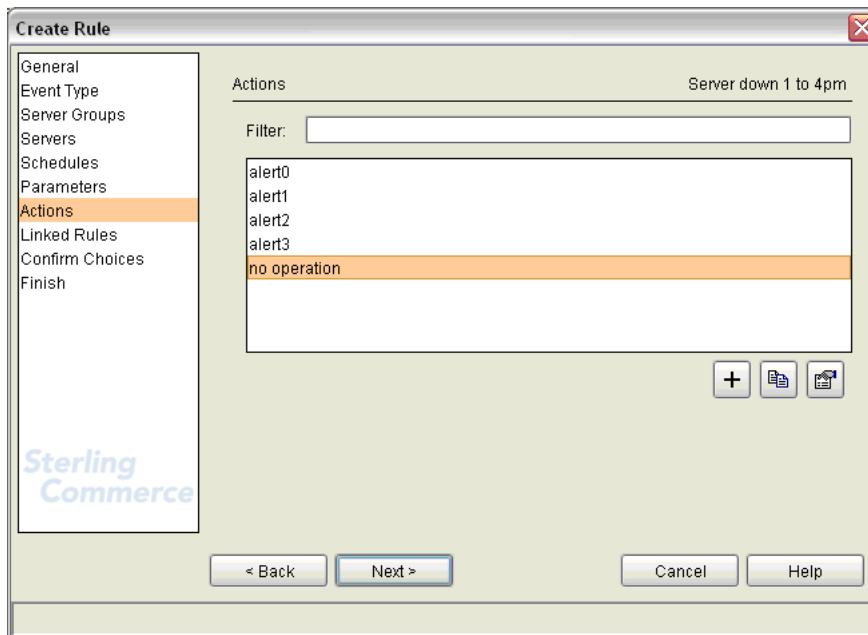


How Can I Build a Schedule To Prevent Rules From Triggering During My Maintenance Windows?

When prompted for a schedule for this new rule, specify the “Sundays 1 to 4pm” rule schedule you previously created.



Change the action to No Operation.



Last, make this second server down rule a higher priority rule than the first server down rule that has no calendar associated with it.

Now, when a server goes down, the Sterling Control Center engine will first see your new server down rule. If it happens to be Sunday between 1 and 4pm, this rule will be triggered and perform no operation. If it is not Sunday between 1 and 4pm, this rule will not be triggered, but the old server

down rule, the one with no associated schedule, will be triggered. Whatever action you specified for this second server down rule will be performed.

By taking advantage of the fact that only one Sterling Control Center rule will be triggered per event, and by putting first the server down rule that takes no action, you accomplish your goal of knowing when a server down event occurs outside of the maintenance window.

Reports Example

The following example illustrates a common use of the Automated Reports facility of Sterling Control Center.

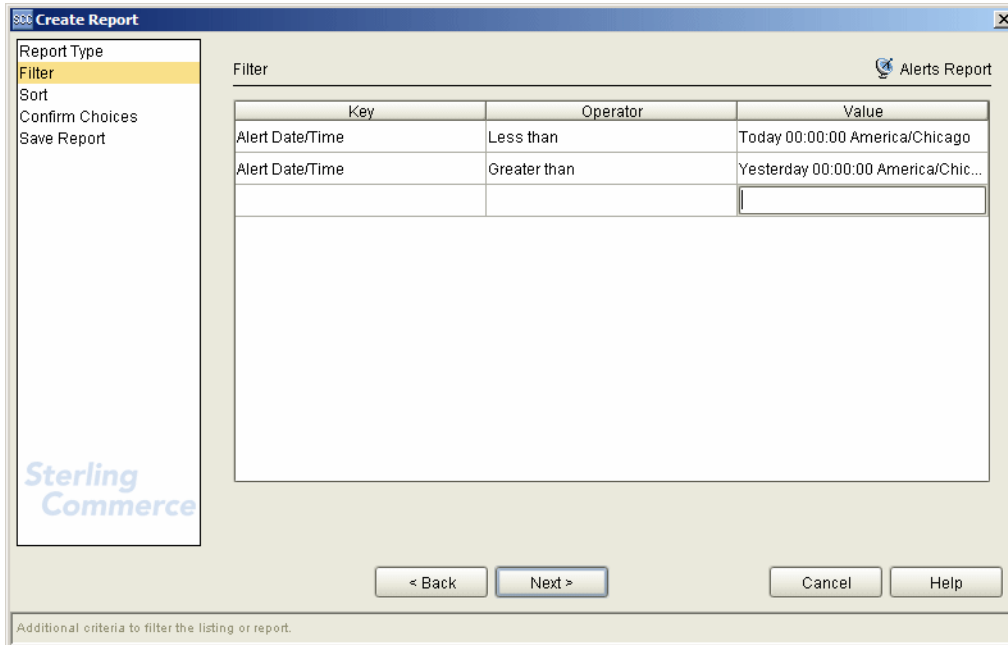
How Do I Set Up an Automated Report of the Previous Day's Alerts?

The basic steps to setting up an automated report that lists all the previous day's alerts are as follows.

- Define an Alerts report and save it.
- Define a report schedule and save it.
- Automate the Alerts report.

Here are the specific steps:

1. Click **Tools > Reports > Define/Run**.
2. Add a new report, select **System** as the Report Category and **Alerts Report** as the Report Type.
3. Select the filter criteria shown below. This narrows the alerts to yesterday's only. This is the key step in this process.



Key	Operator	Value
Alert Date/Time	Less than	Today 00:00:00 America/Chicago
Alert Date/Time	Greater than	Yesterday 00:00:00 America/Chic...

4. Run the report and save it.
5. Define a report schedule by which to run the report (**Tools > Reports > Define Schedule**); for example: Daily at 6:00 AM. Specify the schedule name, choose a calendar (for this example, Daily with a time of 6:00 AM), and save the schedule.
6. Add a new automated report (**Tools > Reports > Automate**).
7. Give the new automated report a name and select the report you defined in step 1 through step 4.
8. Give the automated report the schedule you defined in step 5.
9. Select an e-mail list of recipients to send the mail to.
10. Select other parameters, including the report file format, the file extension, a From address and an mail Subject line, and then save the report. The report is generated daily at the time you specified and sent to the list of addressees you specified.

Set Up SLCs

This chapter offers pointers to help in understanding service-level criteria (SLCs) and how they are implemented. It describes the differences between rules and SLCs, gives tips for deciding when to use one versus the other, and gives examples of common scenarios in which you might want to set up an SLC.

Not all of these examples describe the detailed steps necessary to set up an SLC. Some describe only the idea behind a solution to a problem. For a detailed explanation of SLCs and how to set them up, consult the Help system or the *Sterling Control Center System Administration Guide*.

The following examples are provided:

- What's the Correlation Between SLCs and Rules?
- When Should an SLC Be Used Versus a Rule?
- When Do I Need SLCs?
- What Events May Be Generated By SLCs and When?
- How Do I Know Which Type of SLC to Use?
- How Can I Be Alerted If One Process Runs Too Long?
- How Can I Be Alerted If a Critical Process Fails to Complete On Time?
- How Can I Be Alerted When Related Connect:Enterprise Batches Arrive?
- How Can I Be Alerted If a Related Group of Processes Runs Too Long?
- How Can I Be Alerted If a Related Group of Critical Processes Don't All Complete On Time?
- How Can I Be Alerted If a Critical Process Doesn't Start On Time?
- How Can I Be Alerted If One Or More Related Processes Do Not Start or Complete On Time?
- How Can I Be Alerted If One Or More Related Processes May Not Complete On Time?
- How Can I Reduce the Number of Events with the Same Message Generated By Workflow SLCs?
- How Can I Be Alerted If A Connect:Direct File Transfer May Not Complete On Time?
- How Can I Be Alerted If an Hourly Process Isn't On Time?
- How Can a Producer Know When a File Was Not Delivered On Time?
- How Can a Producer Get Notified if the Consumer Does Not Pick Up a File Within 15 Minutes?
- How Can I Use RUNBATCH To Create Schedules for an Hourly Process?

What's the Correlation Between SLCs and Rules?

SLCs generate SLC type events when certain conditions occur or fail to occur because Processes either execute or fail to execute. Processes can include the following items:

- Connect:Direct Processes
- Sterling Integrator business process activities
- FTP gets and puts
- The movement of data into and out of Connect:Enterprise mailboxes
- Sterling File Gateway Arrived Files, Routes, and Deliveries

The conditions under which SLCs generate events are determined by the SLCs you have defined and enabled.

However, actions (such as generating an alert or sending an e-mail) are performed when rules are triggered. Rules are triggered by events that Sterling Control Center generates either for external reasons (data is collected from managed servers) or for internal reasons (an SLC triggers).

When Should an SLC Be Used Versus a Rule?

You do not use SLCs to the exclusion of rules or vice versa. In fact, without rules SLCs cannot alert you to anything. This is because SLCs do not generate alerts! An SLC may cause an SLC type event to be generated. That event may trigger a rule, which in turn may have an associated action that generates an alert.

The question is: “Which do I need to solve my problem?”

Note: By default, Sterling Control Center comes with a number of predefined, enabled rules. Alerts may be generated simply by defining and enabling SLCs, but only because of these predefined rules. Once you understand what the predefined rules do, you may decide to disable some. Disable rather than delete them as they are not easily restored.

When Do I Need SLCs?

Rules are triggered only by events. SLCs are able to generate events when a specified condition, such as that a Process must start by 10AM, does or does not occur. Always use an SLC when you want to know that something has not happened.

An SLC can tell you when certain things failed to occur, such as:

- Processes or Process steps that are supposed to start by a specific time
- Processes or Process steps that are supposed to end by a specific time
- Processes that are supposed to run for a specified duration

What Events May Be Generated By SLCs and When?

All SLC-related events are of Event Type SLC. They all contain a message ID that indicates why they were generated and the SLC name (groupID), as well as Process information such as Process Name and number when possible.

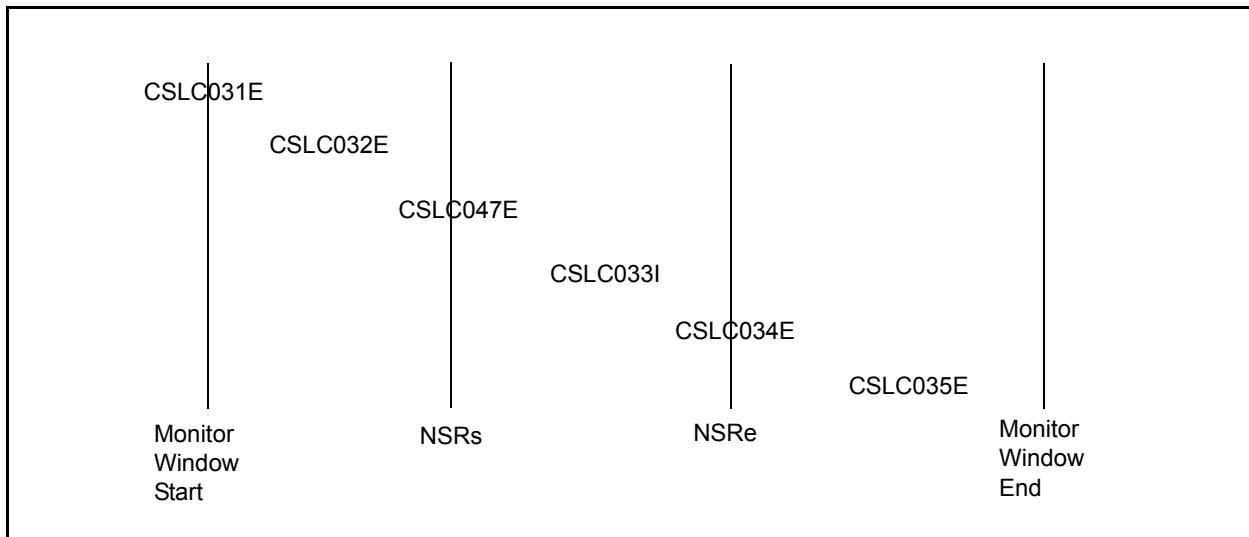
SLC Notifications for Calendar Schedule-Based SLCs

You use a calendar schedule in an SLC when the exact time an occurrence is expected to start or finish is important. Calendar schedules let you specify the normal start range and normal end range for an occurrence. In the charts below:

- NSRs means Normal Start Range start
- NSRe means Normal Start Range end
- NERs means Normal End Range start
- NERe means Normal End Range end

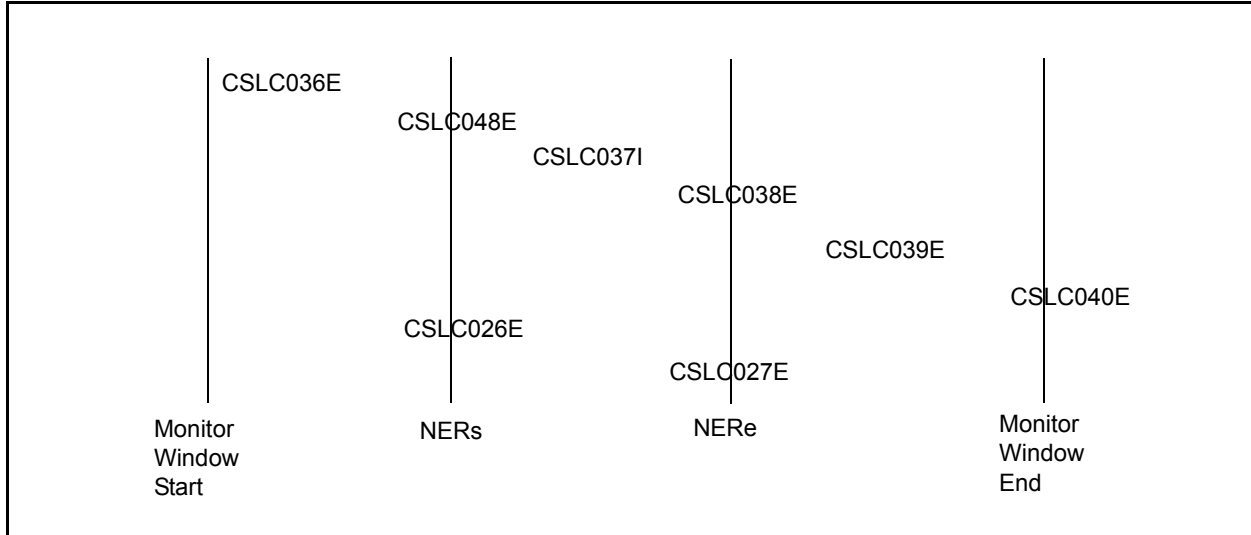
SLC Message IDs Associated with Start Range

The following message IDs are relevant to the start range of an SLC associated with a calendar schedule.



SLC Message IDs Associated with End Range

The following message IDs are relevant to the end range of an SLC associated with a calendar schedule.



Message IDs for Calendar Schedule-Based SLCs

The following message IDs are relevant to an SLC based on a calendar schedule.

Message ID	Message
CSLC026E	Did not start by NERs.
CSLC027E	Did not start by NERe.
CSLC031E	Running prior to start of monitoring
CSLC032E	Started prior to NSRs
CSLC033I	Started.
CSLC034E	Did not start by NSRe.
CSLC035E	Started after NSRe.
CSLC036E	Ended before NERs.
CSLC037I	Ended.
CSLC038E	Did not end by NERe.
CSLC039E	Ended after NERe.
CSLC040E	Did not complete before the end of the monitoring window.
CSLC047E	Did not start by NSRs.
CSLC048E	Running after NERs.

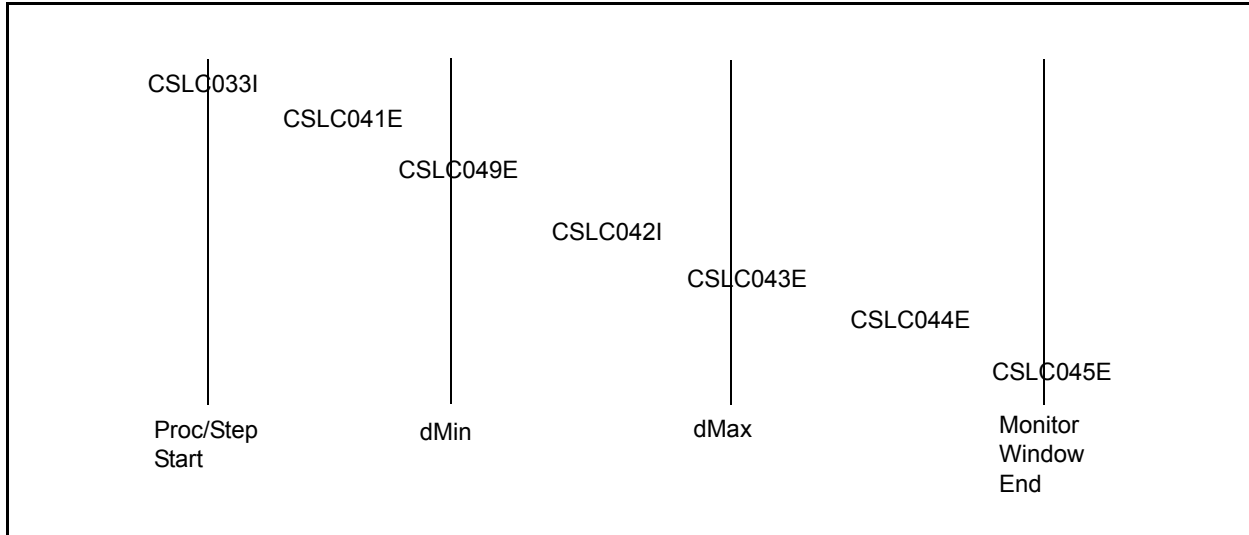
Note: Message IDs CSLC026E and CLSC027E are not generated for SLCs with calendar schedules that include an NSR specification.

SLC Notifications for Duration Schedule-Based SLCs

In the chart below:

- dMin means minimum duration
- dMax means maximum duration

The chart shows when a message ID may come into play for an SLC based on a duration schedule.



Message IDs for Duration Schedule-Based SLCs

These message IDs pertain to schedule-based SLCs.

Message ID	Message
CSLC033I	Started.
CSLC041E	Ended before dMin.
CSLC042I	Ended when expected - between dMin and dMax.
CSLC043E	Did not end by dMax.
CSLC044E	Ended after dMax.
CSLC045E	Did not complete before the end of the monitoring window.
CSLC049E	Did not end by dMin.
CSLC050E	Duration could not be determined - was running before monitoring started.

Other Message IDs for Both Calendar and Duration Schedule-Based SLCs

These message IDs pertain to both duration schedule-based SLCs and calendar schedule-based SLCs.

Message ID	Message
CSLC228E	Did not start before the end of the monitoring window.
CSLC229E	On-time completion may be in jeopardy.
CSLC171I	Step ahead of schedule (25% dMin).
CSLC172E	Step behind schedule (25% dMin).
CSLC173I	Step ahead of schedule (50% dMin).
CSLC174E	Step behind schedule (50% dMin).
CSLC175I	Step ahead of schedule (75% dMin).
CSLC176E	Step behind schedule (75% dMin).
CSLC177I	Step ahead of schedule (25% dMax).
CSLC178E	Step behind schedule (25% dMax).
CSLC179I	Step ahead of schedule (50% dMax).
CSLC180E	Step behind schedule (50% dMax).
CSLC181I	Step ahead of schedule (75% dMax).
CSLC182E	Step behind schedule (75% dMax).

How Do I Know Which Type of SLC to Use?

There are three types of SLCs in Control Center:

- Standard SLCs
- Wildcard SLCs
- Workflow SLCs

If you are interested in knowing when a single Process or Process step runs or fails to run, use either a standard SLC or wildcard SLC. With wildcard SLCs you specify a generic name (using a wildcard character) rather than a specific, individual one. So, if you know the specific Process name, server name, etc., use a standard SLC.

Note: You can use a wildcard SLC in place of a standard SLC, but the Control Center engine processes standard SLCs more efficiently. Use standard SLCs when you have the choice.

If you want to ensure that multiple Processes and/or Process steps run, use a workflow SLC.

How Can I Be Alerted If One Process Runs Too Long?

Since in this situation you're interested only in a single Process, either a standard or wildcard SLC would be appropriate to use. If you know the specific Process and server name, using a standard SLC is more efficient than a wildcard SLC.

Because you're interested only in knowing whether the Process runs for the correct amount of time—not whether it runs during a specific time period—you should define an SLC duration schedule:

The screenshot shows a 'Create SLC Schedule' dialog box with the following fields and options:

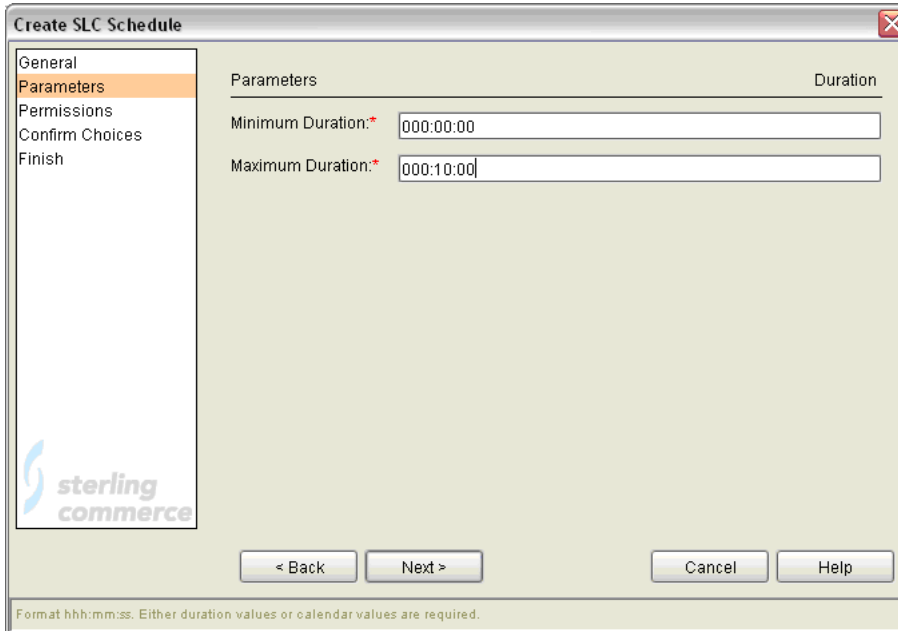
- General** (selected tab)
- Schedule Type:** Duration Schedule
- Name:** Duration
- Description:** (empty)
- Enabled

Buttons: < Back, Next >, Cancel, Help

Footer: Up to 25 characters. Special characters not allowed. Required.

Specify a Maximum Duration value that corresponds to the maximum Process duration.

Specify a Minimum Duration value that corresponds to the minimum Process duration.



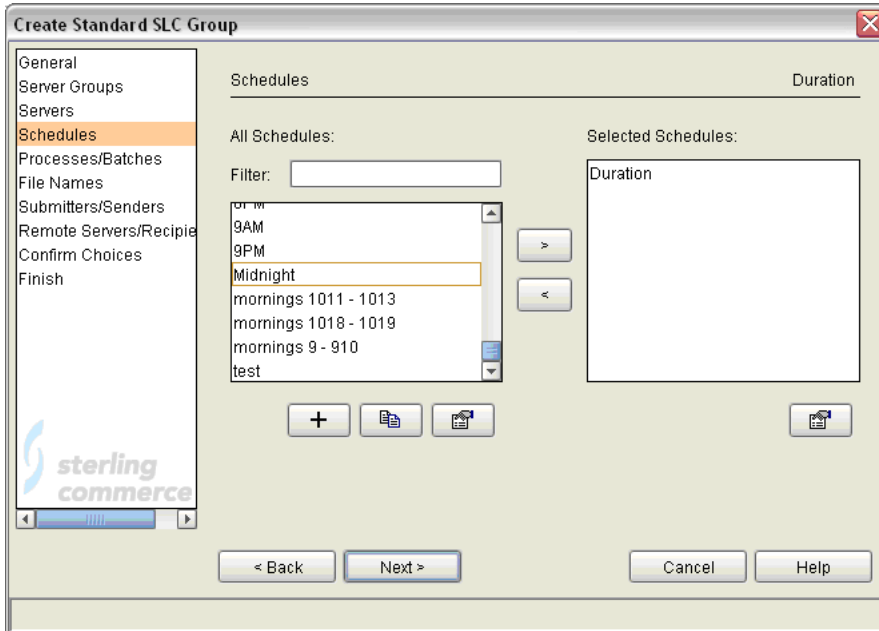
The screenshot shows a dialog box titled "Create SLC Schedule" with a sidebar on the left containing the following items: General, Parameters (highlighted), Permissions, Confirm Choices, and Finish. The main area is titled "Parameters" and "Duration" and contains two input fields: "Minimum Duration:*" with the value "000:00:00" and "Maximum Duration:*" with the value "000:10:00". At the bottom, there are four buttons: "< Back", "Next >", "Cancel", and "Help". A footer note reads: "Format hhh:mm:ss. Either duration values or calendar values are required."

In the above screen shot, the duration schedule specifies that 0 seconds to 10 minutes elapsed time is acceptable for the SLC you associate the schedule with.

Note: Choose the minimum duration wisely for your SLC. Remember that the minimum granularity for measuring the elapsed time of Processes and Process steps in Sterling Control Center is seconds. So, if a Process or Process step runs in less than a second, it may appear to Sterling Control Center to run in 0 seconds.

Sterling Control Center generates events if a Process or Process step takes less than the specified minimum duration. Specifically, an SLC event with message ID CSLC041E is generated. Remember, Processes or Process steps taking less time to run than expected may signal a problem as serious as those that take more than the maximum expected duration. For example, if you are expecting to receive a 100 GB file and the file takes less than a second to be received, there is a good chance you did not receive the expected data.

Reference the duration schedule you have defined when constructing your SLC:



Beyond specifying the duration schedule for your standard or wildcard SLC, specify appropriate values for server groups, servers, and other SLC match criteria to define the Process, Processes, Process step, or Process steps you are interested in knowing about.

How Can I Be Alerted If a Critical Process Fails to Complete On Time?

Use a standard or wildcard SLC and specify appropriate normal end range start (NERS) and normal end range end (NERe) values in the calendar SLC schedule associated with the SLC.

How Can I Be Alerted When Related Connect:Enterprise Batches Arrive?

To be alerted when related batches have arrived in a Connect:Enterprise mailbox, use a workflow SLC. Associate a duration SLC schedule with the workflow SLC with a minimum duration of 00:00:00 and an appropriate maximum duration, which should be the expected time between the arrival of the first and last of the related batches.

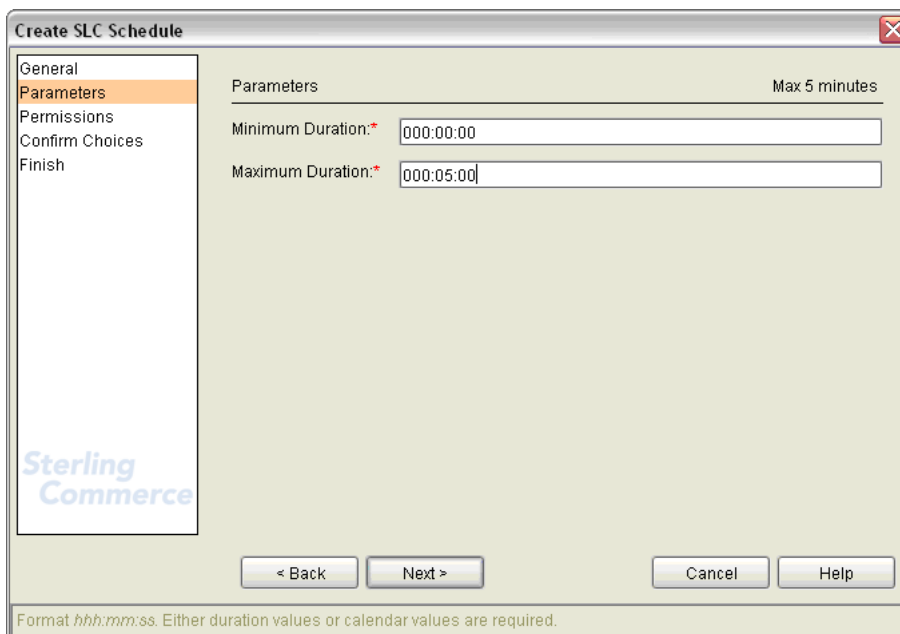
Within the workflow SLC, define one milestone per batch to be received. The milestones you define must specify parameters that uniquely identify each batch.

Each milestone must specify values for duration. Specify minimum and maximum values that match the values in the duration schedule.

Example

Four batches, named File1, File2, File3, and File4 are expected to be received by a managed Connect:Enterprise server named kukla35 within 5 minutes of one another. This data is to be sent by a user named Raj. You want to know when they have all arrived, or that they did not arrive within the allotted time.

First, create a duration SLC schedule to be referenced by the workflow SLC:



The screenshot shows a 'Create SLC Schedule' dialog box with a sidebar on the left containing 'General', 'Parameters', 'Permissions', 'Confirm Choices', and 'Finish'. The 'Parameters' tab is selected. The main area is titled 'Parameters' and 'Max 5 minutes'. It contains two input fields: 'Minimum Duration:*' with the value '000:00:00' and 'Maximum Duration:*' with the value '000:05:00'. At the bottom, there are buttons for '< Back', 'Next >', 'Cancel', and 'Help'. A footer note states: 'Format hh:mm:ss. Either duration values or calendar values are required.'

Because we expect all four batches to arrive within 5 minutes of one another, this duration schedule specifies a dMin of 0 hours, 0 minutes, and 0 seconds, and a dMax of 0 hours, 5 minutes, and 0 seconds.

Next, create the Workflow SLC:

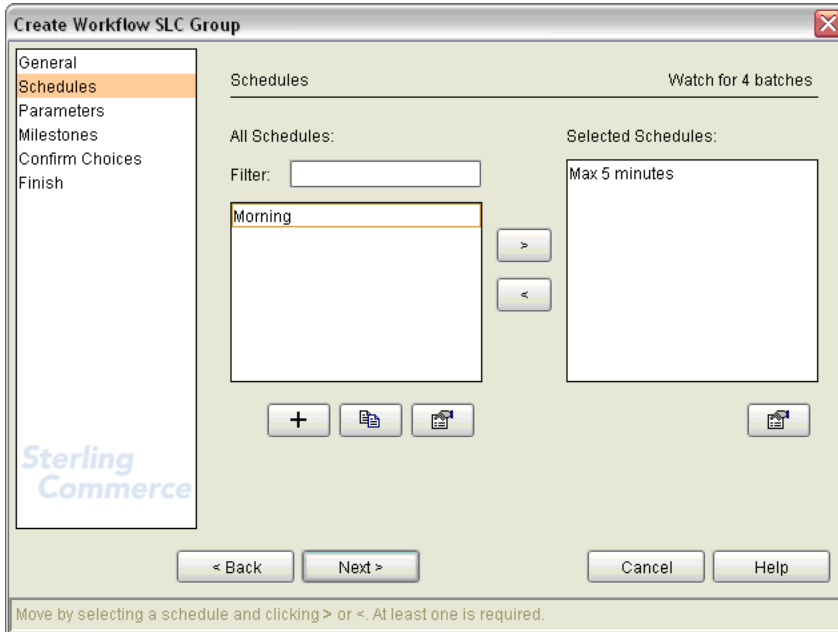
The screenshot shows the 'Create Workflow SLC Group' dialog box with the following configuration:

- Name:** Watch for 4 batches
- Description:** (Empty)
- Enabled:**
- Generate notification if event has not occurred:**
- Suppress Milestone Messages:**
- Monitor Window Tolerances:**
 - Start:** 1 hours
 - End:** 1 hours

Notice the checkbox for Suppress Milestone Messages is selected. This is because we want events generated only for the workflow, not for its milestones.

Also, the End Monitor Window Tolerance is set to 1 hour. This causes an event to be generated if all batches are not received within 1 hour and 5 minutes after the first batch arrives. The event will state that the workflow did not complete within the monitoring window.

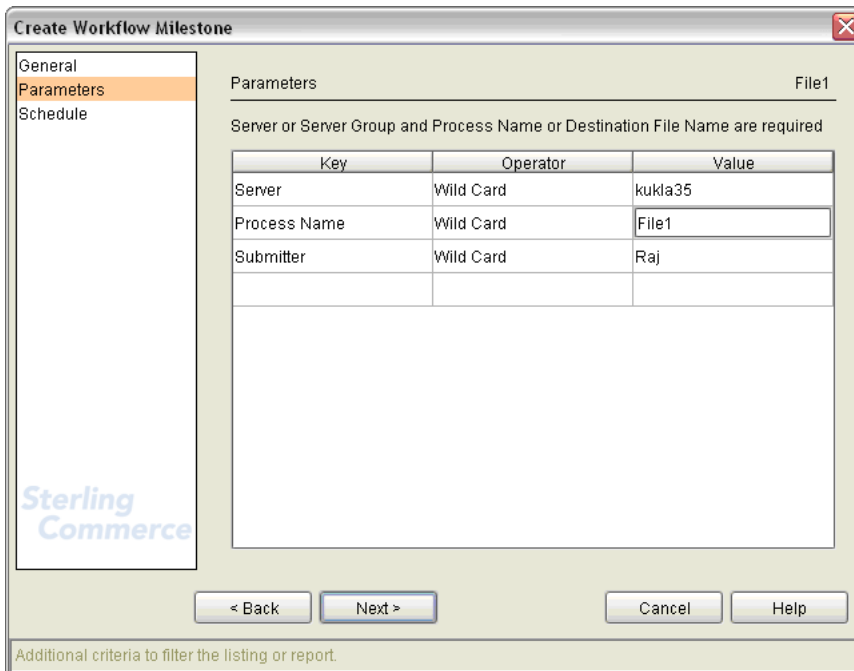
Now select the duration schedule created in a previous step:



The concurrence count must be set to the number of related groups of batches expected to arrive per hour. We expect at most one per hour, so set concurrence count to 1.

Next, add the first milestone to the workflow SLC, the one corresponding to the first batch.

Here are the parameters for the milestone, named File1:




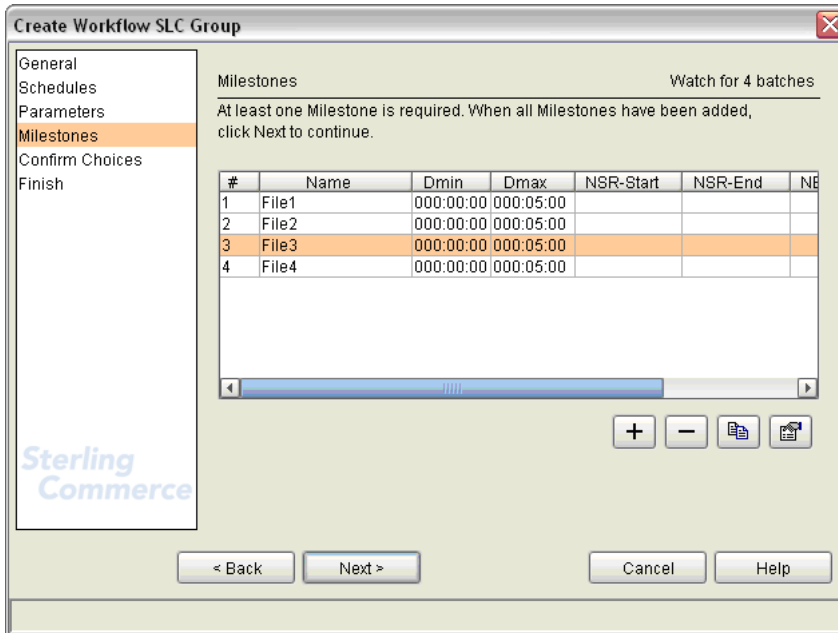
Note that the server the batches are to arrive on is specified, as is the name of the batch and the sender of the data. Remember for Connect:Enterprise, Process name equates to batch ID, and submitter equates to the origin of the data.

Now set the schedule for the milestone:

The screenshot shows a dialog box titled "Create Workflow Milestone" with a close button in the top right corner. On the left, there is a vertical navigation pane with three tabs: "General", "Parameters", and "Schedule". The "Schedule" tab is selected and highlighted in orange. The main area of the dialog is titled "Schedule" and contains several input fields. Under the "Duration:" section, there are two text boxes: "Minimum Duration(Dmin):" with the value "000:00:00" and "Maximum Duration(Dmax):" with the value "000:05:00". Below this is a "Time Range:" section with four text boxes: "Normal Start Range (NSR) - Start:", "Normal Start Range (NSR) - End:", "Normal End Range (NER) - Start:", and "Normal End Range (NER) - End:". At the bottom left of the main area is the "Sterling Commerce" logo. At the bottom right are five buttons: "< Back", "Next >", "Finish", "Cancel", and "Help". A status bar at the very bottom of the dialog contains the text: "Maximum Duration. Format hh:mm:ss. Either duration values or calendar values are required."

The minimum and maximum durations correspond to the durations for the workflow SLC schedule.

Now all you need to do is duplicate this milestone three times using the Duplicate button . Duplicate the milestone once for each additional related batch, setting the Process name to the new related batch name.



Control Center comes with built-in rules that watch for corresponding SLC events that potentially could be generated as a result of this workflow SLC. They are:

- Proc duration short - CSLC041E - alert2
- Proc duration okay - CSLC042I - alert0
- Proc duration not ended - CSLC043E - alert2
- Proc duration exceeded - CSLC044E - alert1
- Proc duration way over - CSLC045E - alert1

You may want to add your own rules and actions that are similar to the existing, built-in rules, which reference these message IDs. Do so by duplicating them, giving them new names, and specifying additional parameters, such as the name of the workflow SLC, and specifying your own actions.

For example:

Parameters Proc duration short - watch for 4 batches

Key	Operator	Value
Message Id	Matches	CSLC041E
SLC Name	Matches	Watch for 4 batches

Multiple entries allowed. Optional.

Be sure, after constructing the new rules, to assign them a higher priority than existing rules that specify the same SLC event message ID. More specific rules should always be higher in priority than their more general counterparts.

That's all that needs be done. This workflow SLC will be triggered whenever a batch arrives with an ID of File1, File2, File3, or File4 that was sent by Raj to kukla35. An SLC event will be generated when the last of the four batches arrives, or after 5 minutes, and after an hour and 5 minutes if they do not all arrive.

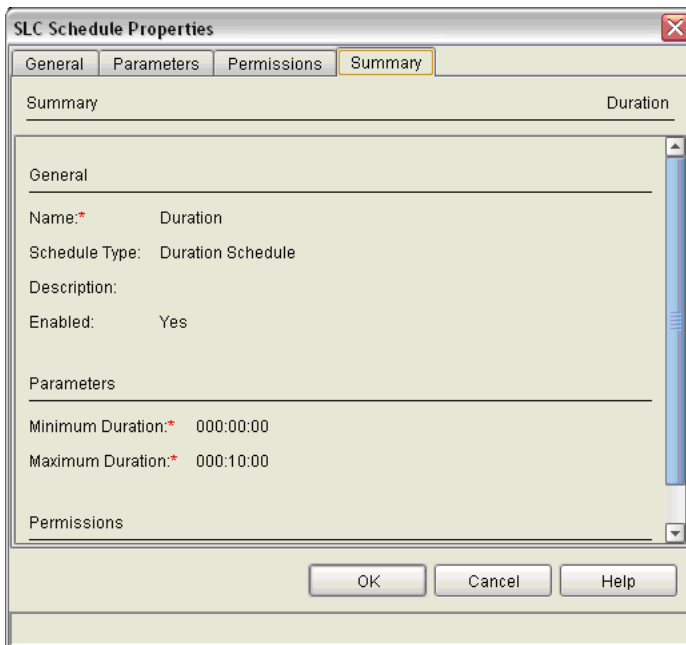
How Can I Be Alerted If a Related Group of Processes Runs Too Long?

To monitor whether a group of Processes runs too long, construct an SLC duration schedule and reference that schedule in a workflow SLC group that contains appropriate values for server groups, servers, and other SLC match criteria that specify the Process, Processes, Process step, or Process steps you're interested in.

For example, say that PROCESSA runs on SERVERA, PROCESSB runs on SERVERB, and PROCESSC runs on SERVERC, and that all must run within 10 minutes.

Since you're interested in knowing that multiple related Processes run within a specified amount of time, you'll use a workflow SLC and associate a duration SLC schedule with it.

For this example, the duration SLC schedule should have the following attributes:



The screenshot shows a dialog box titled "SLC Schedule Properties" with a close button (X) in the top right corner. The dialog has four tabs: "General", "Parameters", "Permissions", and "Summary", with "Summary" selected. The "Summary" tab is divided into two sections: "Summary" and "Duration".

Summary Section:

- General
- Name:* Duration
- Schedule Type: Duration Schedule
- Description:
- Enabled: Yes

Parameters Section:

- Minimum Duration:* 000:00:00
- Maximum Duration:* 000:10:00

Permissions Section:

At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

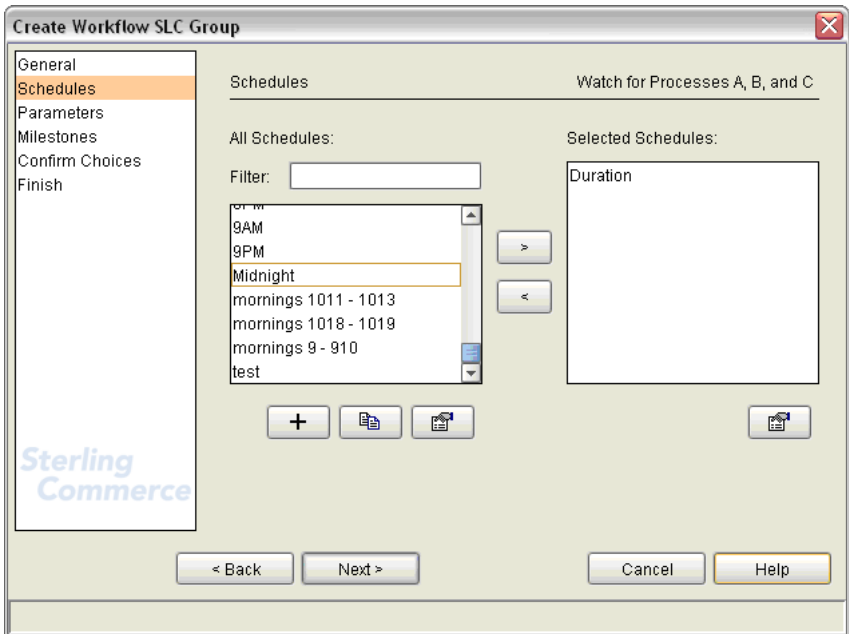
A minimum duration of no time and a maximum duration of 10 minutes are specified.

The workflow SLC that will reference this duration SLC schedule should be defined as follows:

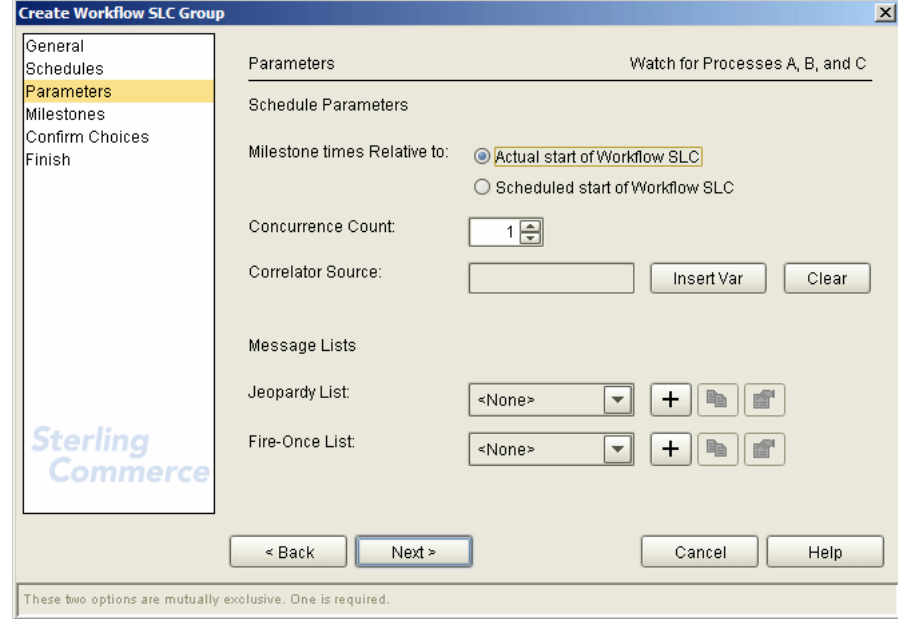
Up to 25 characters. Special characters not allowed. Required.

Monitor Window Tolerances: Start and End specify when you want an SLC to begin and end monitoring. Note that in this instance, because a duration SLC schedule is being associated with the workflow SLC, Start Monitor Window Tolerance is immaterial. That's because we don't care *what time* the Processes begin, only that they all run within a specified duration.

The End Monitor Window Tolerance tells Sterling Control Center how long to continue watching for the end of the workflow beyond the schedule associated with the workflow SLC. You may make the value for both the Start and End Monitor Window Tolerances as short as zero hours.



In the previous screen shot, we have specified the duration SLC schedule previously defined, which specifies a minimum duration of 0 seconds and a maximum duration of 10 minutes.

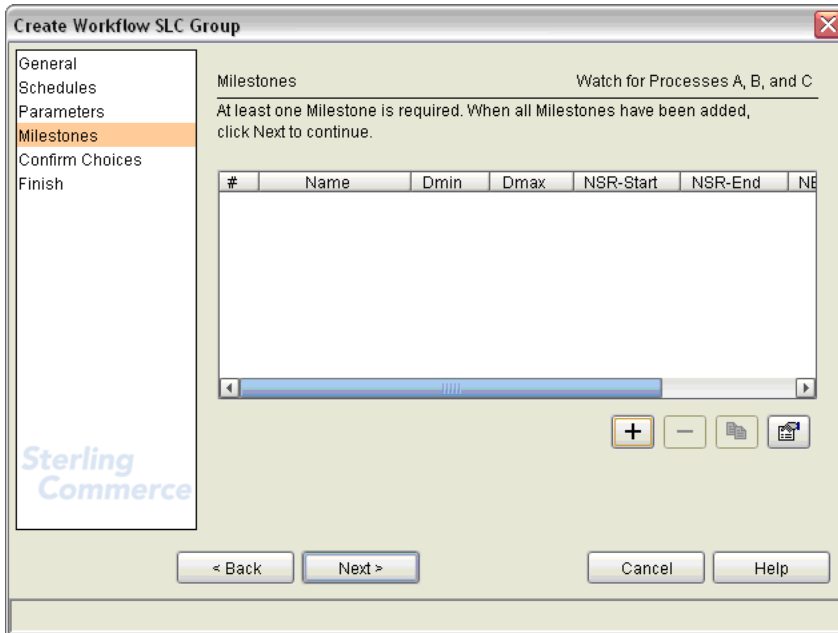


The value you select for Milestone Times Relative To, Actual Start or Scheduled Start, will only matter if NSR and/or NER values are specified for the SLC's milestones.

Concurrency Count is set to 1 because we expect only one occurrence of this workflow to be executing within the monitoring window specified.

Note that, for SLCs with SLC calendar schedules, Concurrency Count tells Control Center how many occurrences of the workflow are expected to run during the scheduled time. If “Generate notification if event has not occurred” is selected for an SLC with a calendar schedule, then SLC events will be generated for each instance that does not run as expected. For SLCs with SLC duration schedules, Concurrency Count tells Control Center how many occurrences of the workflow may run simultaneously.

Next you are presented with the Milestones window:



Specify three milestones, one for each Process (PROCESSA, PROCESSB, and PROCESSC).

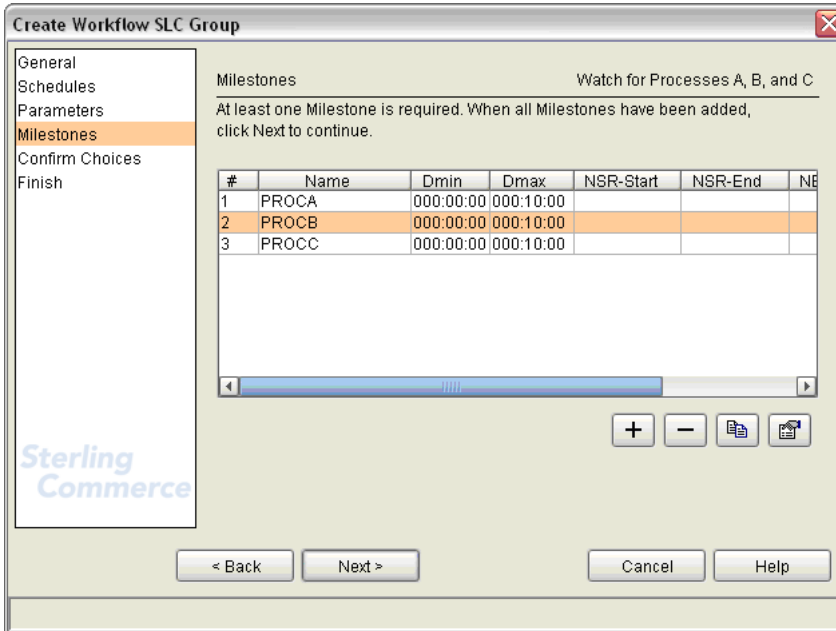
Additional criteria to filter the listing or report.


Define parameters specifying PROCESSA on SERVERA for the first milestone, which we will call PROCA.

Format hhh:mm:ss. Either duration values or calendar values are required.

Specify a minimum and maximum duration of 000:00:00 to 000:10:00 (0 minutes to 10 minutes) for milestone PROCA.

Additional milestones, similar to the first, must be defined for PROCESSB on SERVERB, and PROCESSC on SERVERC.



Tip: Using the Duplicate button () to copy the PROCA milestone makes it easier to create milestones PROCB and PROCC.

How Can I Be Alerted If a Related Group of Critical Processes Don't All Complete On Time?

Use a workflow SLC. Associate a calendar SLC schedule with the workflow with appropriate times for normal end range start (NERs) and normal end range end (NERe).

Within the workflow SLC, define one milestone per critical Process.

The milestones you define must specify parameters to uniquely identify each Process.

Each milestone must specify values for duration, and/or values for NSRs and NSRe, and/or values for NERs and NERe. The choices you make here affect only the additional milestone SLC events that may be generated.

If you know approximately how long each Process in the group should last, specify dMin and dMax values for each milestone. Milestone SLC events are generated if the Processes run shorter than the dMin value specified, or if they run longer than the dMax value specified.

The NSR and NER values are all relative to when the workflow begins, which, if you specify Milestone Times Relative to Actual Start of Workflow SLC, is determined by the start of any one of the Processes corresponding to the milestones defined. If you specify Milestone Times Relative to Scheduled Start of Workflow SLC, NSR and NER values are relative to the scheduled start of the workflow. You may wish to specify NSR values for one or more of the milestones if you know that other Processes within the workflow should begin within a specified amount of time of the workflow itself starting.

Create Workflow SLC Group

Parameters Process Group Not Completed on Time

Schedule Parameters

Milestone times Relative to: Actual start of Workflow SLC
 Scheduled start of Workflow SLC

Concurrency Count:

Correlator Source:

Message Lists

Jeopardy List:

Fire-Once List:

Range = 1-99. Default = 0. Optional.

How Can I Be Alerted If a Critical Process Doesn't Start On Time?

Use a standard or wildcard SLC. Specify NSRs and NSRe values in the calendar SLC schedule associated with the SLC.

How Can I Be Alerted If One Or More Related Processes Do Not Start or Complete On Time?

Use a workflow SLC. Associate a calendar SLC schedule with the workflow with appropriate times for NSRs, NSRe, NERs, and NERe.

Within the workflow SLC, define one milestone per critical Process. The milestones you define must specify parameters to uniquely identify each Process.

Each milestone must specify values for duration, and/or values for NSRs and NSRe, and/or values for NERs and NERe. These choices affect only the additional milestone SLC events that may be generated.

If you know approximately how long each Process in the group should last, specify dMin and dMax values for each milestone. Milestone SLC events will be generated if the Processes run shorter than the dMin value specified, or if they run longer than the dMax value specified.

If you select Milestone Times Relative to Actual Start of Workflow SLC when setting up the workflow, the NSR and NER values for milestones are relative to when the workflow begins, which is when any of the Processes corresponding to the workflow's milestones start.

If you select Milestone Times Relative to Scheduled Start of Workflow SLC when setting up the workflow, the NSR and NER values specified for milestones are relative to the workflow's scheduled start time (its NSRs value in the schedule assigned to it).

You may wish to specify NSR values for one or more of the milestones if you know other Processes within the workflow should begin within a specified amount of time of the workflow itself starting.

How Can I Be Alerted If One Or More Related Processes May Not Complete On Time?

Workflow SLCs have a parameter called Jeopardy List.

The screenshot shows the 'Create Workflow SLC Group' dialog box with the 'Parameters' tab selected. The 'Jeopardy List' field is highlighted, indicating it is the focus of the discussion. The dialog box includes a sidebar with navigation options: General, Schedules, Parameters (selected), Milestones, Confirm Choices, and Finish. The main area contains the following fields and controls:

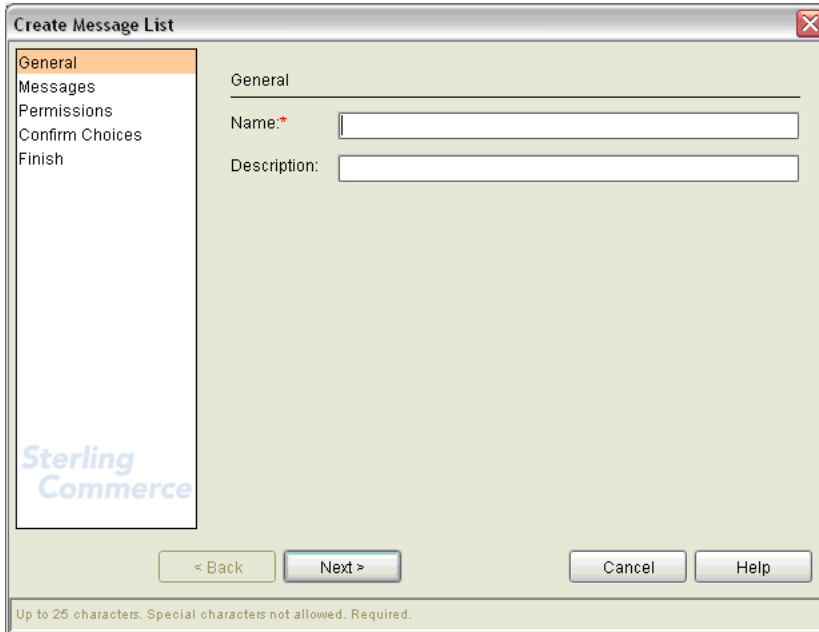
- Parameters: Testwf2
- Schedule Parameters
- Milestone times Relative to: Actual start of Workflow SLC, Scheduled start of Workflow SLC
- Concurrency Count: 4 (spin box)
- Correlator Source: [] [Insert Var] [Clear]
- Message Lists
- Jeopardy List: <None> [] [+] [] []
- Fire-Once List: <None> [] [+] [] []

Buttons at the bottom: < Back, Next >, Cancel, Help. A footer note reads: Range = 1-99, Default = 0, Optional.

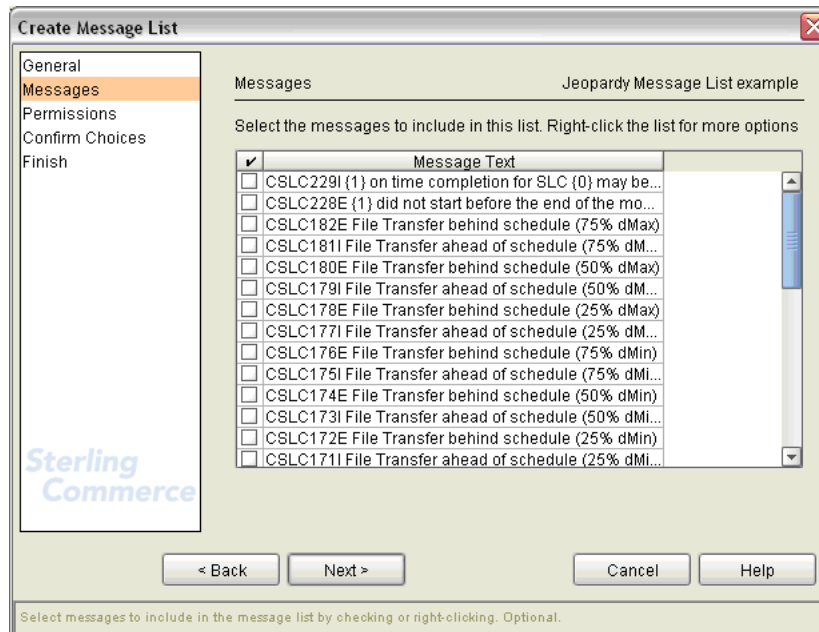
A jeopardy list is simply a list of message IDs that may be associated with SLC events.

For the lists of message identifiers that may be associated with SLC events, see *SLC Notifications for Calendar Schedule-Based SLCs* on page 97 and *SLC Notifications for Duration Schedule-Based SLCs* on page 99.

To create a jeopardy list, press the + button to the right of the Jeopardy List drop-down list:



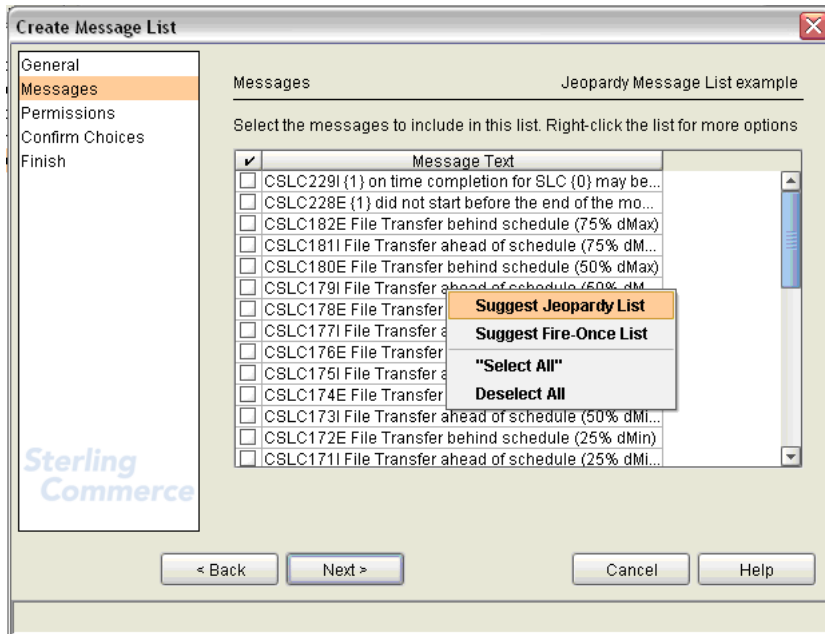
Specify a **Name** and **Description**, then press **Next**.



On the subsequent wizard screen, check the message IDs you want to use as an indication that the workflow may not complete on time. If any SLC events are generated for the workflow that contain a message ID specified in the jeopardy list, then an additional SLC event with a message ID of

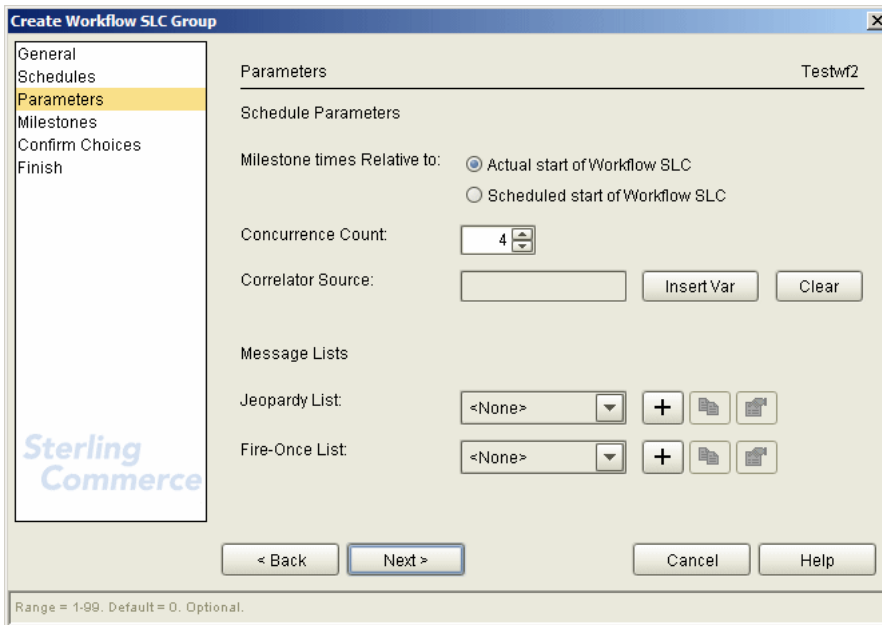
CSLC229I—on time completion in jeopardy—will be generated for every workflow milestone that has yet to start.

Note: Right-click the message list and choose Suggest Jeopardy List to get a suggested list of messages.



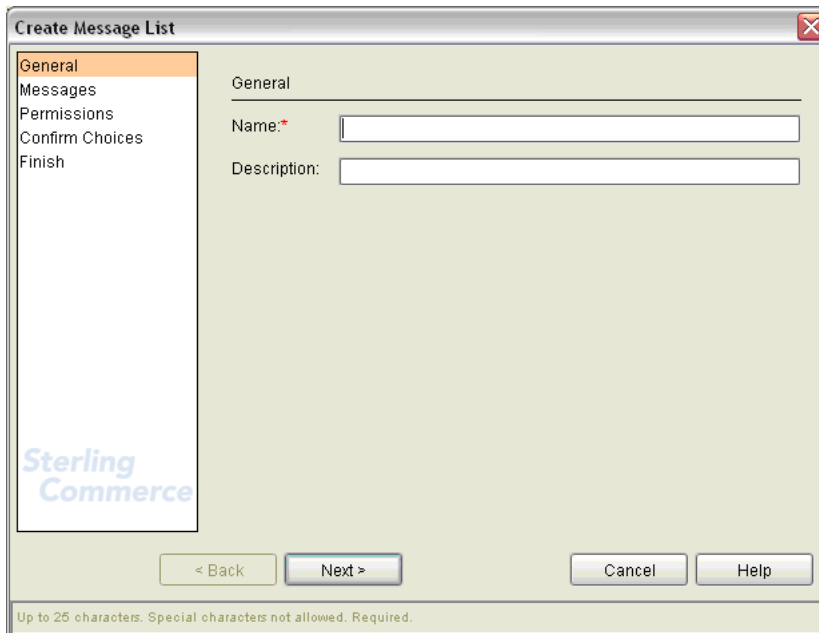
How Can I Reduce the Number of Events with the Same Message Generated By Workflow SLCs?

If one milestone in a workflow SLC does not start on time, you may not care to be notified that every subsequent milestone is also going to be late starting. To suppress these subsequent SLC events, use the Fire-Once List parameter when setting up your workflow SLC:



The screenshot shows the 'Create Workflow SLC Group' dialog box with the 'Parameters' tab selected. The left sidebar contains a tree view with 'Parameters' highlighted. The main area is titled 'Parameters' and 'Testwf2'. It includes a 'Schedule Parameters' section with 'Milestone times Relative to:' set to 'Actual start of Workflow SLC'. The 'Concurrency Count' is set to 4. The 'Correlator Source' field is empty, with 'Insert Var' and 'Clear' buttons. The 'Message Lists' section has 'Jeopardy List' and 'Fire-Once List' both set to '<None>', each with a '+' button and a file icon. At the bottom are '< Back', 'Next >', 'Cancel', and 'Help' buttons. A footer note reads 'Range = 1-99, Default = 0, Optional.'

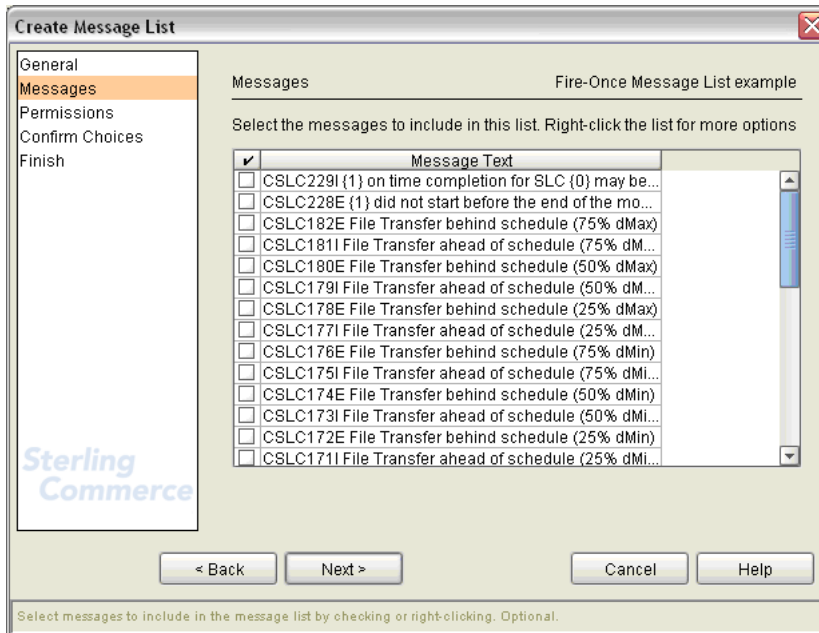
To create a fire-once list, click + to the right of Fire-Once List:



The screenshot shows the 'Create Message List' dialog box with the 'General' tab selected. The left sidebar contains a tree view with 'General' highlighted. The main area is titled 'General' and has 'Name:*' and 'Description:' fields. At the bottom are '< Back', 'Next >', 'Cancel', and 'Help' buttons. A footer note reads 'Up to 25 characters. Special characters not allowed. Required.'

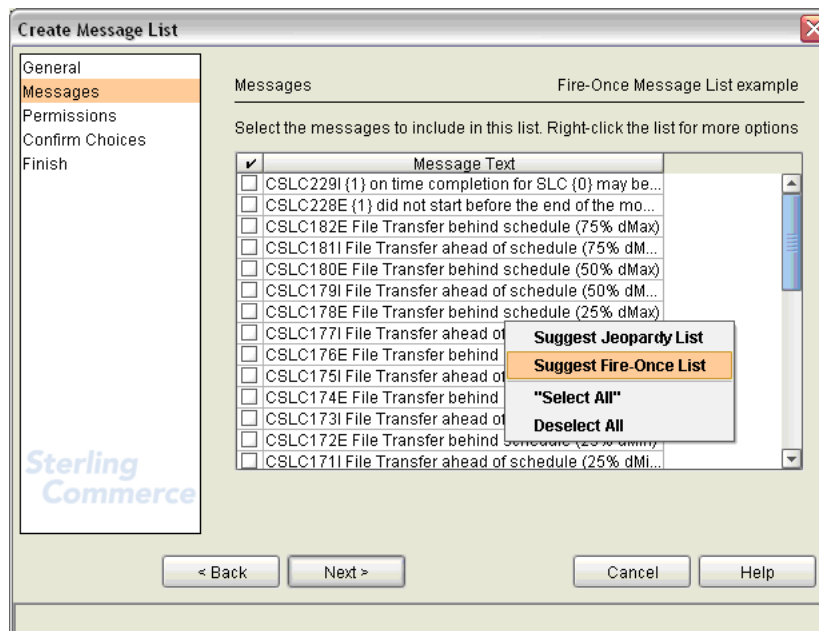
Specify a **Name** and **Description**, then press **Next**.

The following wizard panel is displayed:



Select the message IDs for which you want SLC events to be generated only once per instance of the workflow.

Note: To get a suggested list of fire-once messages, right-click the messages displayed and choose Suggest Fire-Once List.



How Can I Be Alerted If A Connect:Direct File Transfer May Not Complete On Time?

When Sterling Control Center can ascertain the size of a file being transmitted, and you have specified a duration SLC schedule for the transfer, SLC events with the following message IDs may be generated.

Message ID	Message Text
CSLC171I	Step ahead of schedule (25% dMin)
CSLC172E	Step behind schedule (25% dMin)
CSLC173I	Step ahead of schedule (50% dMin)
CSLC174E	Step behind schedule (50% dMin)
CSLC175I	Step ahead of schedule (75% dMin)
CSLC176E	Step behind schedule (75% dMin)
CSLC177I	Step ahead of schedule (25% dMax)
CSLC178E	Step behind schedule (25% dMax)
CSLC179I	Step ahead of schedule (50% dMax)
CSLC180E	Step behind schedule (50% dMax)
CSLC181I	Step ahead of schedule (75% dMax)
CSLC182E	Step behind schedule (75% dMax)

Sterling Control Center comes with predefined rules configured to watch for these events.

Examine the rules in the Rules Listing to determine whether these rules fit your requirements.

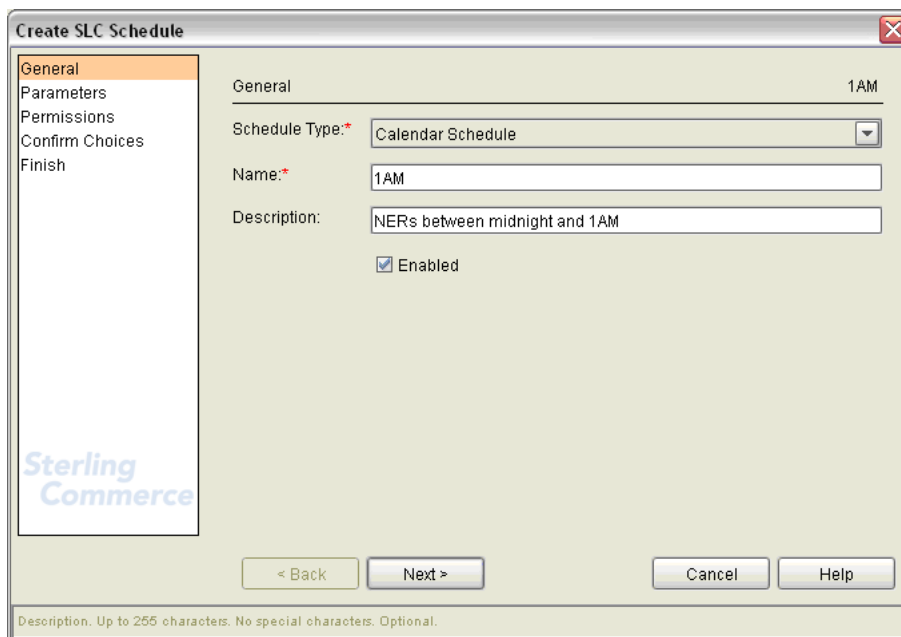
How Can I Be Alerted If an Hourly Process Isn't On Time?

Say you have a Process named HOURLY that is supposed to run every hour on Server REGULAR. How can you be alerted should it fail to run?

The easiest way to handle this requirement is to create 24 different calendar SLC schedules, one for each hour of the day, and then assign all 24 of them to one standard SLC that specifies appropriate matching criteria—a Server name of REGULAR and a Process name of HOURLY—for its parameters.

Select **Manage > Service Level Criteria (SLCs) > SLC Schedules** and click + to create a new schedule.

First, create a schedule with NERs between midnight and 1AM



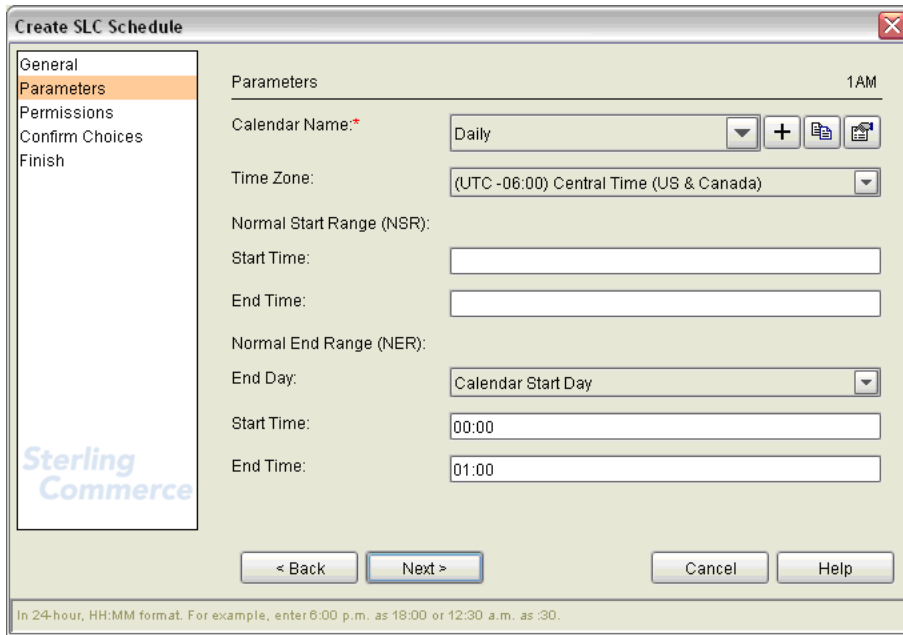
The screenshot shows a 'Create SLC Schedule' dialog box with the following fields and options:

- General** (selected tab)
- General** (header)
- Schedule Type:** Calendar Schedule (dropdown menu)
- Name:** 1AM (text field)
- Description:** NERs between midnight and 1AM (text field)
- Enabled:** Enabled (checkbox)
- Buttons:** < Back, Next >, Cancel, Help
- Footer:** Description. Up to 255 characters. No special characters. Optional.

This schedule will be for the first hour of the day.

As our Process is supposed to run 24 hours a day, 7 days a week, set Calendar Name to the predefined calendar called Daily.

It is not necessary to set a start range. Simply setting an end range encompassing the hour within which the Process is supposed to run is sufficient.

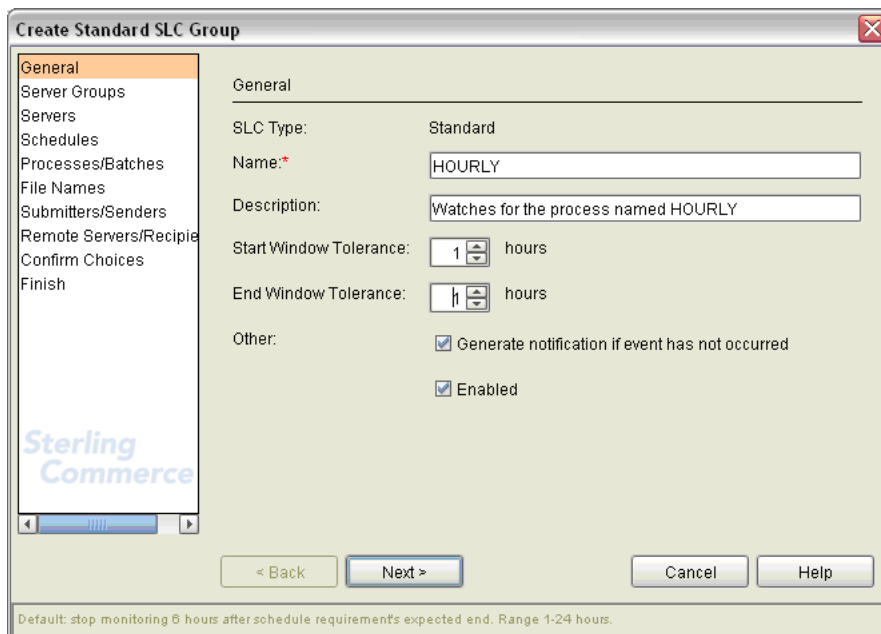


Now duplicate this SLC calendar schedule 23 times, modifying the name and NER values appropriately.

SLC Calendar Schedule Name	NERs	NERe
Midnight to 1AM	00:00	01:00
1AM to 2AM	01:00	02:00
2AM to 3AM	02:00	03:00
3AM to 4AM	03:00	04:00
4AM to 5AM	04:00	05:00
5AM to 6AM	05:00	06:00
6AM to 7AM	06:00	07:00
7AM to 8AM	07:00	08:00
8AM to 9AM	08:00	09:00
9AM to 10AM	09:00	10:00
10AM to 11AM	10:00	11:00
11AM to 12PM	11:00	12:00
12PM to 1PM	12:00	13:00
1PM to 2PM	13:00	14:00

SLC Calendar Schedule Name	NERs	NERe
2PM to 3PM	14:00	15:00
3PM to 4PM	15:00	16:00
4PM to 5PM	16:00	17:00
5PM to 6PM	17:00	18:00
6PM to 7PM	18:00	19:00
7PM to 8PM	19:00	20:00
8PM to 9PM	20:00	21:00
9PM to 10PM	21:00	22:00
10PM to 11PM	22:00	23:00
11PM to Midnight	23:00	24:00

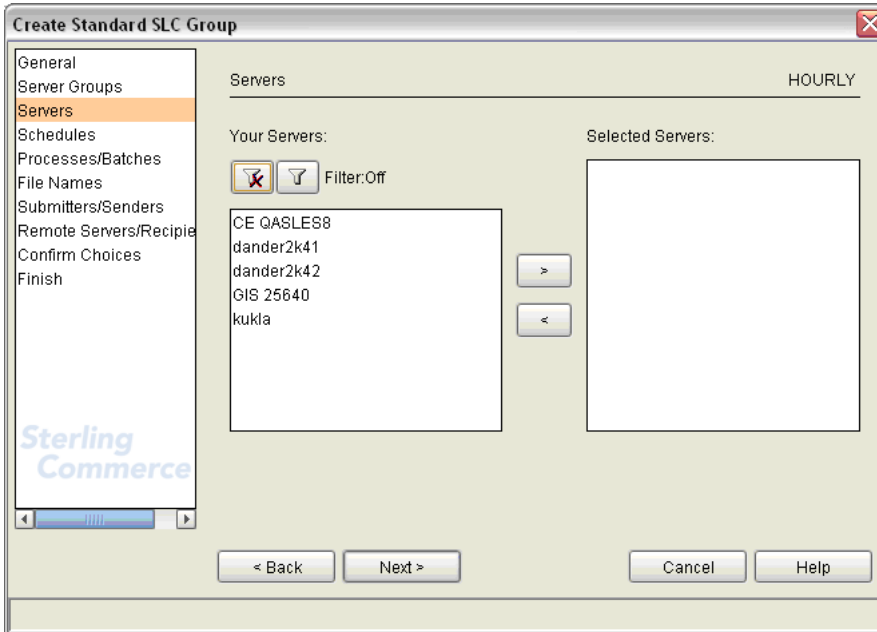
Since you know the exact Process and server names (HOURLY and REGULAR, respectively), use a standard SLC to generate events when your Process, HOURLY, does not run.



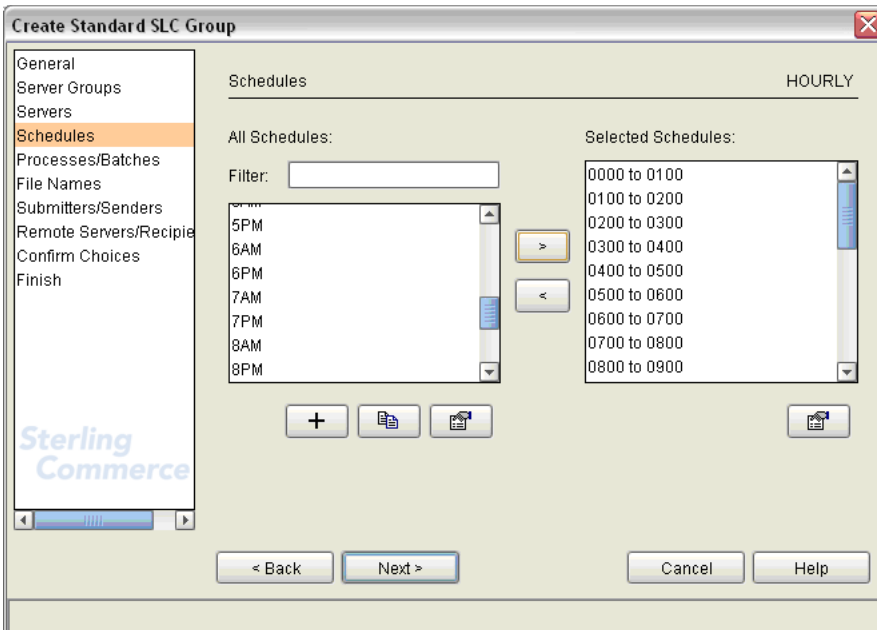
Set the Start and End Window Tolerance values to 1 hour. This allows the hourly Process to start or end as much as one hour early or late and still be counted as running. If you do not want any tolerance at all, you may set the Start and End Window Tolerance values to zero hours.

Check Generate Notification if Event Has Not Occurred to cause SLC events to be generated when HOURLY does not run.

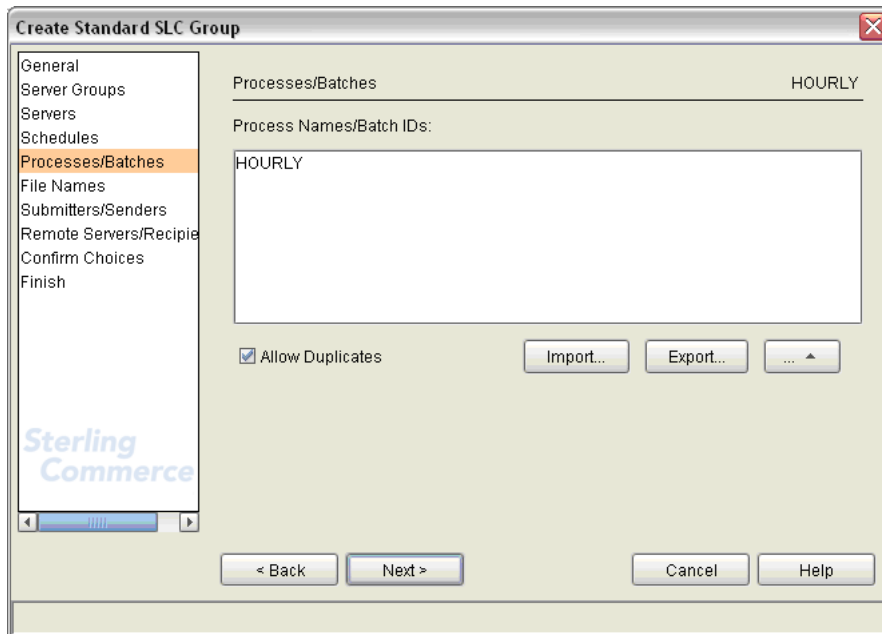
Select the Server REGULAR



Select all 24 calendar SLC schedules you have created:



Specify Process name HOURLY:



How Can a Producer Know When a File Was Not Delivered On Time?

Set up a workflow SLC that uses a calendar schedule to define a typical time range with a normal end range start (NERs) and normal end range end (NERe). In addition to the workflow schedule and other required information, enter the following parameters for the Workflow SLC's milestone substituting appropriate values for the keys, Server, SI.Mailbox Path, and Dest File Name, that meet your needs:

The screenshot shows the 'Update Workflow Milestone' dialog box with the 'Parameters' tab selected. The dialog has three tabs: 'General', 'Parameters', and 'Schedule'. The 'Parameters' tab is active, showing a table of parameters for milestone 'ms1'. A message above the table states: 'Server or Server Group and Process Name or Destination File Name are required'. The table has three columns: 'Key', 'Operator', and 'Value'. Below the table is a 'Correlator Source' field with 'Insert Var' and 'Clear' buttons. At the bottom are 'Update', 'Cancel', and 'Help' buttons. The status bar at the bottom left reads 'Milestone Parameters: Key:'.

Key	Operator	Value
Server	Wild Card	*
SI.Is Put(true/false)	Wild Card	true
SI.Mailbox Path	Wild Card	/Producer/Mailbox
Dest File Name	Wild Card	Producerfile.txt

Correlator Source:

Milestone Parameters: Key:

How Can a Producer Know When a File Was Not Delivered On Time?

To define the milestone schedule, specify the maximum amount of time for the system to wait before a milestone SLC event is generated.

The screenshot shows a dialog box titled "Update Workflow Milestone" with a close button (X) in the top right corner. The dialog has three tabs: "General", "Parameters", and "Schedule", with "Schedule" selected. The "Schedule" tab contains the following fields and controls:

- A label "Schedule" followed by the text "ms1".
- A section labeled "Duration:" containing two input fields:
 - "Minimum Duration(Dmin):" with the value "000:00:00".
 - "Maximum Duration(Dmax):" with the value "000:15:00".
- A section labeled "Time Range:" containing four input fields:
 - "Normal Start Range (NSR) - Start:"
 - "Normal Start Range (NSR) - End:"
 - "Normal End Range (NER) - Start:"
 - "Normal End Range (NER) - End:"
- At the bottom, three buttons: "Update", "Cancel", and "Help".
- A footer note: "In 24-hour, hh:mm format. Ex: enter 6:00 p.m. as 18:00 or 12:30 a.m. as :30."

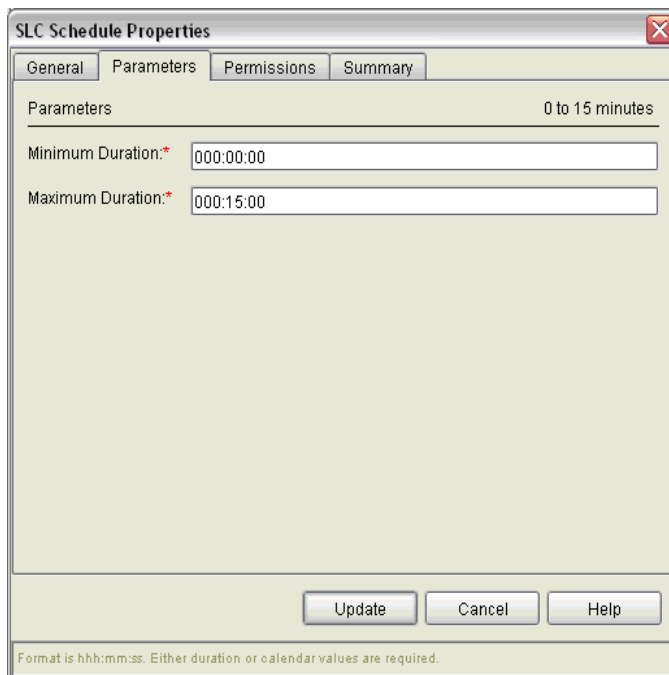
How Can a Producer Get Notified if the Consumer Does Not Pick Up a File Within 15 Minutes?

To notify a producer if the consumer does not pick up a file within 15 minutes of its arrival, use a Workflow SLC with a duration schedule and two milestones:

- Milestone 1 to watch for the delivery (called *File delivered to consumer MBX* in this example)
- Milestone 2 to monitor the pickup of the data delivered (called *Consumer Picks Up the File*)

Since multiple files can be picked up and delivered simultaneously, you must also specify a correlator value for the Workflow, which will tie the name of the file being delivered to the name of the file being picked up.

First, create a workflow SLC with a duration schedule of 15 minutes by setting the minimum and maximum duration values:



The screenshot shows a dialog box titled "SLC Schedule Properties" with a close button (X) in the top right corner. The dialog has four tabs: "General", "Parameters", "Permissions", and "Summary". The "Parameters" tab is selected. The "Parameters" section is titled "Parameters" and has a subtitle "0 to 15 minutes". Below this, there are two input fields: "Minimum Duration:*" with the value "000:00:00" and "Maximum Duration:*" with the value "000:15:00". At the bottom of the dialog, there are three buttons: "Update", "Cancel", and "Help". A footer note at the bottom of the dialog reads: "Format is hhh:mm:ss. Either duration or calendar values are required."

How Can a Producer Get Notified if the Consumer Does Not Pick Up a File Within 15 Minutes?

To ensure that the milestones within the workflow are associated with the same file, specify destFile as the Correlator Source at the workflow level:

The screenshot shows the 'Create Workflow SLC Group' dialog box with the 'Parameters' tab selected. The 'Parameters' section is titled 'Pickup within 15 minutes'. The 'Milestone times Relative to:' section has two radio buttons: 'Actual start of Workflow SLC' (selected) and 'Scheduled start of Workflow SLC'. The 'Concurrency Count' is set to 1. The 'Correlator Source' is 'destFile', with 'Insert Var' and 'Clear' buttons. The 'Message Lists' section has 'Jeopardy List' and 'Fire-Once List' both set to '<None>', with '+' and document icons. The 'Sterling Commerce' logo is visible in the bottom left. Navigation buttons '< Back', 'Next >', 'Cancel', and 'Help' are at the bottom.

To create the first milestone to watch for deliveries, specify the following parameters substituting appropriate values for the keys, SI.Mailbox Path, Dest File Name, and Server, that meet your needs:

The screenshot shows the 'Update Workflow Milestone' dialog box with the 'Parameters' tab selected. The 'Parameters' section is titled 'File delivered to consumer MBX'. A note states: 'Server or Server Group and Process Name or Destination File Name are required'. A table lists the parameters:

Key	Operator	Value
SI.Mailbox Path	Wild Card	/RQ103PGPCConsumerE/Inbox*
Dest File Name	Wild Card	*
Server	Wild Card	*
SI.Is Put(true/false)	Wild Card	true

The 'Correlator Source' is empty, with 'Insert Var' and 'Clear' buttons. Navigation buttons 'Update', 'Cancel', and 'Help' are at the bottom. The footer text reads 'Milestone Parameters: Key:'.

To define the milestone schedule, again specify 15 minutes for the duration.

The screenshot shows a 'Create Workflow Milestone' dialog box with the following fields and values:

- Duration:** (Label)
- Minimum Duration(Dmin):** 00:00:00
- Maximum Duration(Dmax):** 00:15:00
- Time Range:** (Label)
- Normal Start Range (NSR) - Start:** (Empty field)
- Normal Start Range (NSR) - End:** (Empty field)
- Normal End Range (NER) - Start:** (Empty field)
- Normal End Range (NER) - End:** (Empty field)

Buttons at the bottom: < Back, Next >, Finish, Cancel, Help.

Footer text: Format hhh:mm:ss. Either duration or calendar values required.

To create the second milestone to watch for pickups that do not occur within 15 minutes of file delivery (SI.Is Put is false), specify the following criteria as parameter keys:

Key	Operator	Value
SI.Mailbox Path	Wild Card	The pickup location, for example, /RQ103PGPConsumerE/Inbox*
Dest File Name	Wild Card	*
SI.Is Put(true/false)	Wild Card	false
Server	Wild Card	*

How Can a Producer Get Notified if the Consumer Does Not Pick Up a File Within 15 Minutes?

The screenshot shows the 'Update Workflow Milestone' dialog box with the 'Parameters' tab selected. The title bar reads 'Update Workflow Milestone' and the subtitle is 'Consumer picks up the file'. Below the subtitle, it states 'Server or Server Group and Process Name or Destination File Name are required'. A table with three columns: 'Key', 'Operator', and 'Value' contains the following data:

Key	Operator	Value
SI.Mailbox Path	Wild Card	/RQ103PGPConsumerE\Inbox*
Dest File Name	Wild Card	*
SI.Is Put(true/false)	Wild Card	false
Server	Wild Card	*

Below the table is a 'Correlator Source:' text box, followed by 'Insert Var' and 'Clear' buttons. At the bottom are 'Update', 'Cancel', and 'Help' buttons. A status bar at the very bottom reads 'Milestone Parameters: Key.'

As the last step, define the schedule for the second milestone the same way you did for the first by specifying 15 minutes for the duration.

The screenshot shows the 'Update Workflow Milestone' dialog box with the 'Schedule' tab selected. The title bar reads 'Update Workflow Milestone' and the subtitle is 'Consumer picks up the file'. Under the 'Schedule' section, there are several input fields:

- 'Duration:' section:
 - Minimum Duration(Dmin): 000:00:00
 - Maximum Duration(Dmax): 000:15:00
- 'Time Range:' section:
 - Normal Start Range (NSR) - Start: [empty]
 - Normal Start Range (NSR) - End: [empty]
 - Normal End Range (NER) - Start: [empty]
 - Normal End Range (NER) - End: [empty]

At the bottom are 'Update', 'Cancel', and 'Help' buttons. A status bar at the very bottom reads 'Format is hhh:mm:ss. Either duration or calendar values are required.'

How Can I Use RUNBATCH To Create Schedules for an Hourly Process?

RUNBATCH takes a template and a script. See the *Creating Multiple Objects* appendix in the *Sterling Control Center System Administration Guide* for more information about the RUNBATCH utility and how to use it.

Template

The template name is `recurringschedule.tmp`. It is located in the Sterling Control Center *installation directory/conf/templates* directory:

```
<calendarSchedule>
  <id>&name;</id>
  <desc>&name;</desc>
  <enabled>true</enabled>
  <ner>
    <start>&start;</start>
    <end>&end;</end>
    <day>0</day>
  </ner>
  <calendarId>Daily</calendarId>
  <timeZone>America/Chicago</timeZone>
  <roles></roles>
</calendarSchedule>
```

Note: Substitute an appropriate value for `timeZone` in your template.

Script

The script name is `recurringschedulescript.txt`. It should be put in the Sterling Control Center *installation directory/conf* directory.

How Can I Use RUNBATCH To Create Schedules for an Hourly Process?

```
copy recurringschedule
&name;=0000 to 0100
&start;=00:00
&end;=01:00
SCHEDULE
copy recurringschedule
&name;=0100 to 0200
&start;=01:00
&end;=02:00
SCHEDULE
copy recurringschedule
&name;=0200 to 0300
&start;=02:00
&end;=03:00
SCHEDULE
copy recurringschedule
&name;=0300 to 0400
&start;=03:00
&end;=04:00
SCHEDULE
copy recurringschedule
&name;=0400 to 0500
&start;=04:00
&end;=05:00
SCHEDULE
copy recurringschedule
&name;=0500 to 0600
&start;=05:00
&end;=06:00
SCHEDULE
copy recurringschedule
&name;=0600 to 0700
&start;=06:00
&end;=07:00
SCHEDULE
copy recurringschedule
&name;=0700 to 0800
&start;=07:00
&end;=08:00
SCHEDULE
copy recurringschedule
&name;=0800 to 0900
&start;=08:00
&end;=09:00
SCHEDULE
copy recurringschedule
&name;=0900 to 1000
&start;=09:00
&end;=10:00
```

(Continued...)

```
SCHEDULE
copy recurringschedule
&name;=1000 to 1100
&start;=10:00
&end;=11:00
SCHEDULE
copy recurringschedule
&name;=1100 to 1200
&start;=11:00
&end;=12:00
SCHEDULE
copy recurringschedule
&name;=1200 to 1300
&start;=12:00
&end;=13:00
SCHEDULE
copy recurringschedule
&name;=1300 to 1400
&start;=13:00
&end;=14:00
SCHEDULE
copy recurringschedule
&name;=1400 to 1500
&start;=14:00
&end;=15:00
SCHEDULE
copy recurringschedule
&name;=1500 to 1600
&start;=15:00
&end;=16:00
SCHEDULE
copy recurringschedule
&name;=1600 to 1700
&start;=16:00
&end;=17:00
SCHEDULE
copy recurringschedule
&name;=1700 to 1800
&start;=17:00
&end;=18:00
SCHEDULE
copy recurringschedule
&name;=1800 to 1900
&start;=18:00
&end;=19:00
SCHEDULE
copy recurringschedule
&name;=1900 to 2000
&start;=19:00
&end;=20:00
SCHEDULE
copy recurringschedule
&name;=2000 to 2100
&start;=20:00
&end;=21:00
```

(Continued...)

How Can I Use RUNBATCH To Create Schedules for an Hourly Process?

```
SCHEDULE
copy recurringschedule
&name;=2100 to 2200
&start;=21:00
&end;=22:00
SCHEDULE
copy recurringschedule
&name;=2200 to 2300
&start;=22:00
&end;=23:00
SCHEDULE
copy recurringschedule
&name;=2300 to 2400
&start;=23:00
&end;=24:00
SCHEDULE
```

Troubleshoot SLCs

Troubleshoot SLCs

SLCs can fail to work as expected for many reasons. Here we define “working” as generating appropriate SLC events at the times determined by the SLC criteria, the schedule or schedules associated with the SLC, and the Process type events that occurred or did not occur on managed servers specified by the SLC. This chapter discusses examples of SLC events that were not generated as expected and why. It also discusses how to generate an SLC Debug Report to assist you in troubleshooting SLC issues.

Some reasons that SLC events fail to be generated include the following:

- SLC events are not generated by an SLC if it is not enabled.
- SLC events are not generated if Process type events—Process Started, Process Ended, Process Step Started, and Process Step Ended—do not match the SLC parameters specified.
- Back-level versions of managed servers, that is, managed servers not on current maintenance, may not generate statistics records, or may generate statistics records without the needed information, for the events constructed by Sterling Control Center to have the information needed to match the SLC parameters specified.

SLC Tips

Here are some tips to follow when dealing with SLCs:

- To prevent extraneous events being generated by the SLC service, define only two of the following items in the schedules associated with an SLC:
 - Normal Start Range (NSR)
 - Normal End Range (NER)
 - Duration

That is, if an SLC has a calendar schedule that specifies both an NSR and NER, do not associate a duration schedule with the SLC too. If an SLC has a duration schedule associated with it, and you also want to associate a calendar schedule with it, the calendar schedule should specify either an NSR or an NER, but not both. Do this to avoid conflicts that would occur due to the inherent duration derived by taking the difference between the calendar schedule NSR and NER values and the explicit duration specified in the duration schedule.

- Review the predefined rules Sterling Control Center comes with for SLC event types, and the actions they specify. Ensure that only the rules you wish to be triggered are enabled and that the actions they specify are appropriate. Disable rules that watch for events you do not care about.

For example, Sterling Control Center ships with a rule named Process Not Started Warn, which is triggered when an SLC event is generated at the start of the start range, NSRs, specified if an appropriate Process type event—Process started, or Process step started—does not transpire by that time. If knowing that a Process or Process step has not started by NSRs is immaterial to you, either disable the Process Not Started Warn rule—best choice—or change the action that it refers to no operation—next best choice—or change the action's Alert Severity to <no alert generated>.

How Do I Fix My SLC?

If SLCs would do what you meant, as opposed to what you told them to do, they would work as expected the first time, but sometimes that is not the case. There are many ways to set up SLCs incorrectly.

Here are things to look for when your SLCs don't do what you intended:

SLC Not Enabled

It may be obvious, but sometimes SLCs do not work properly simply because they are not enabled. Always check SLCs that do not function correctly and make sure they are enabled.

Create Wildcard SLC Group

General

Server Groups

Servers

Schedules

Processes/Batch IDs

Destination File Names

Submitters/Mailbox IDs

Remote Servers

Confirm Choices

Finish

General

SLC Type: Wildcard

Name: Test

Description:

Start Window Tolerance: 6 hours

End Window Tolerance: 6 hours

Concurrency Count: 1

Other:

Generate notification if event has not occurred

Enabled

< Back Next > Cancel Help

Check to enable the wildcard SLC. Optional.

Match Criteria

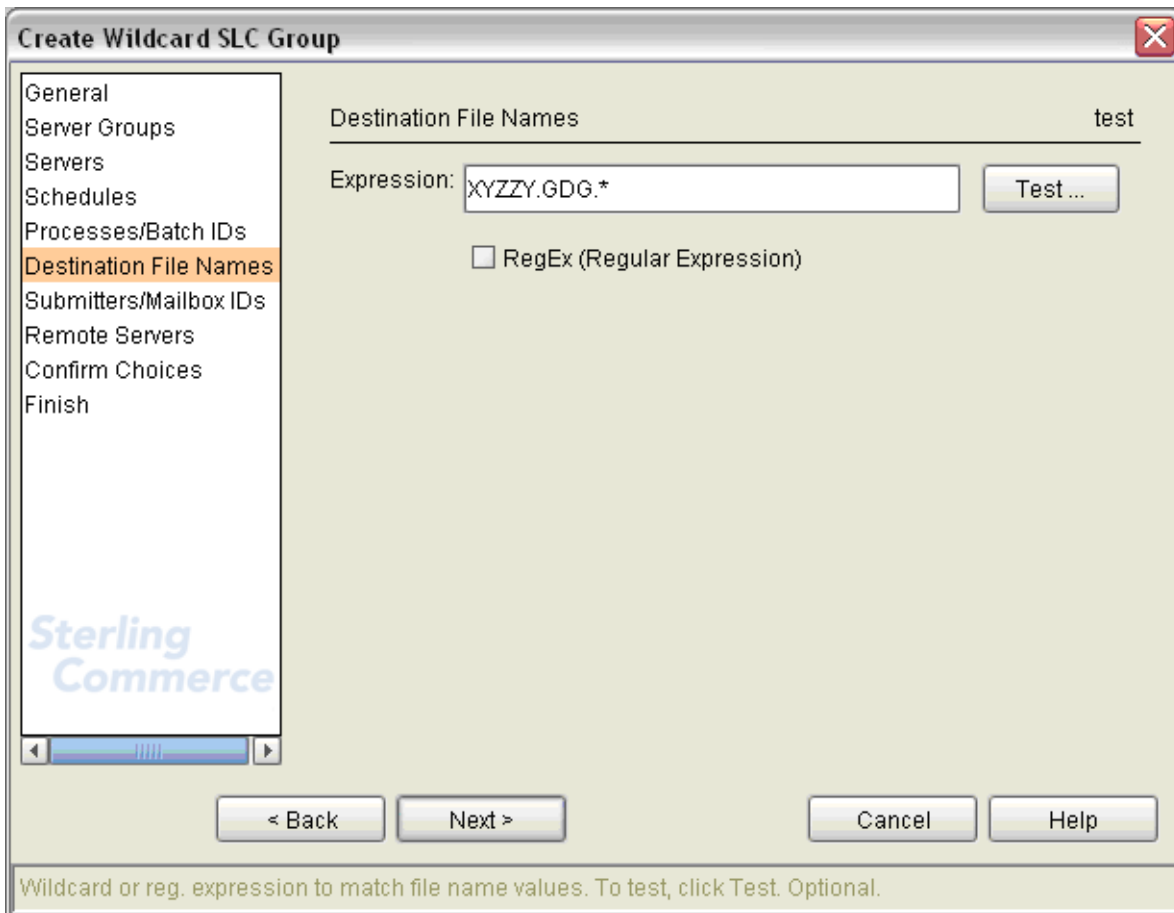
The primary reason SLCs fail to work as planned (assuming they are enabled) is that the data in the events that should trigger them does not exactly match the criteria specified.

For example, you may specify a Process name in the SLC of Xyzzy, but if the Process name is actually XYZZY, it will not trigger the SLC. Why? The *yzzy* in the criteria is in lower case while the actual name shows *YZZY* in upper case, which does not match.

Or, you may specify a destination file name of *c:\myfile.dat*. But if the destination file name is actually *C:\myfile.dat*, it will not trigger the SLC. Why? The *c:* in the criteria is in lower case, while the actual name shows *C:* in upper case, which does not match.

There are times when you must either use a wildcard SLC or a workflow SLC, as opposed to a standard SLC, to accomplish your monitoring goals. One of those times is you when you are monitoring transfers to generation data groups (GDGs). When transferring files to GDGs, the destination file name is not known until the file is actually created.

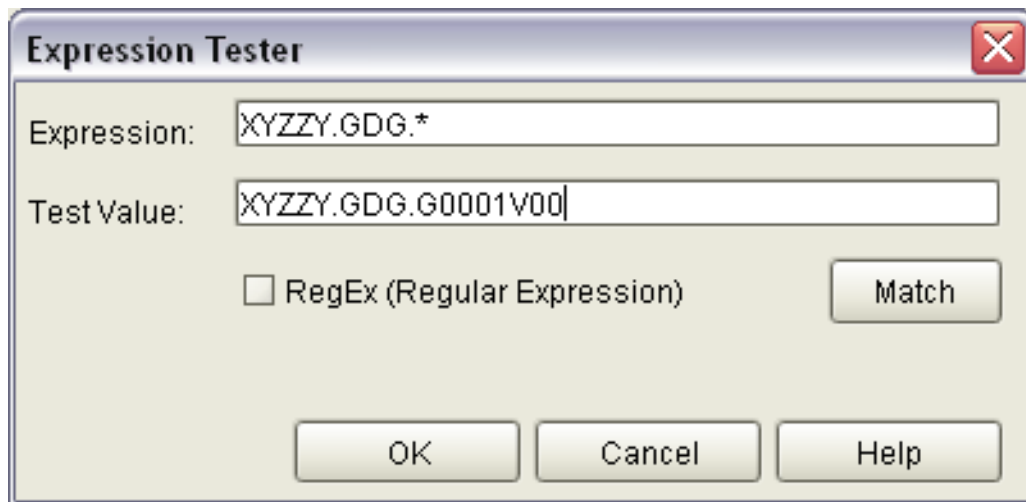
For example, if your process is transferring a file to the dataset *XYZZY.GDG(+1)*, the SLC destination file name should be specified as *XYZZY.GDG.** (assuming the Regular Expression check box is not set).



Note: When specifying match criteria for wildcard SLCs or milestones in workflow SLCs, use only the Regex (Regular Expression) option if you have specified a regular expression for the match criteria.

To find out whether specifying faulty match criteria is the reason your SLC is not working properly, run a Database Events report to view details of the events you expect to trigger your SLC. Specify event parameters that match the Process you feel should have triggered your SLC but did not. Include the XML_STRING value at a minimum in the report output, then closely compare the match criteria specified in your SLC with the output generated.

The Test button on many SLC wizard panels is also helpful. Use it to try out your wildcard or regular expressions to see if they match the values you expect the events that trigger your SLC to contain.



A.M. versus P.M.

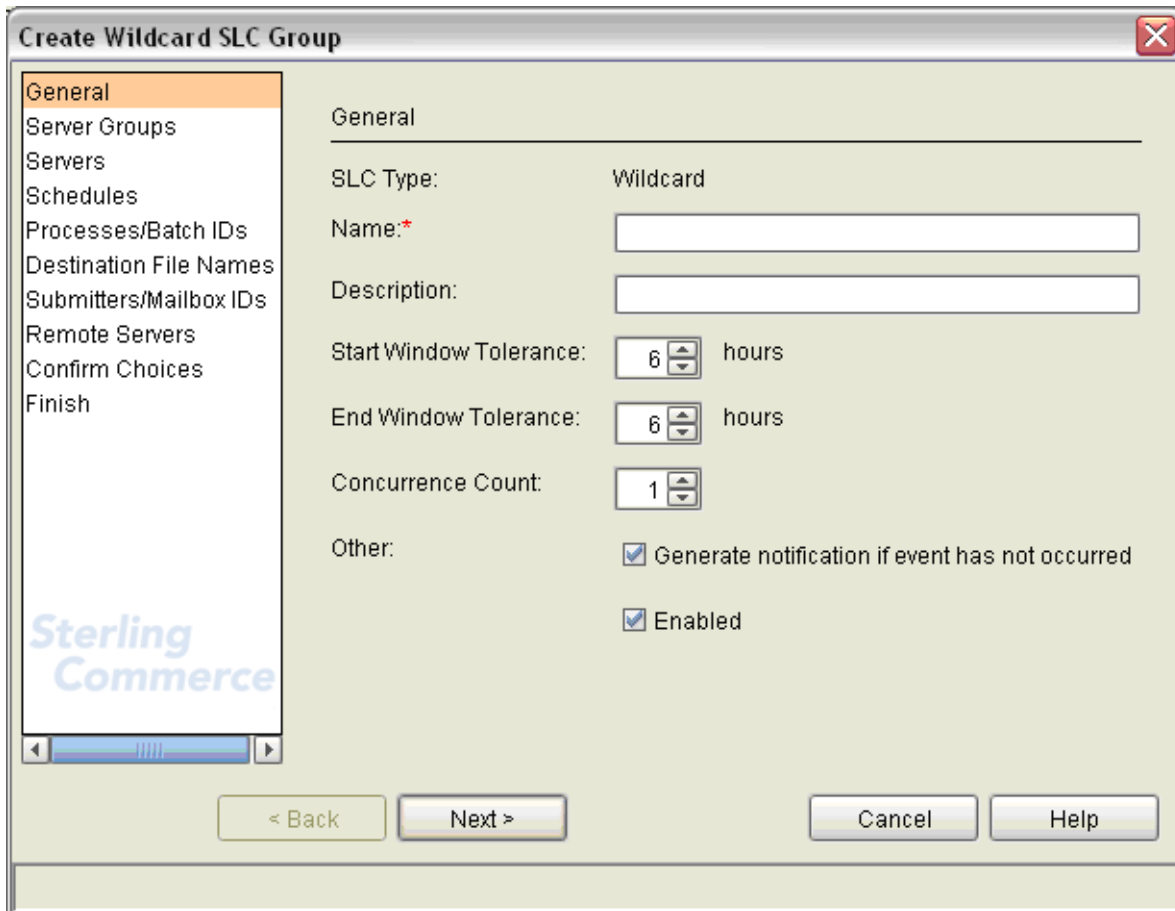
When constructing calendar schedules for use with SLCs, or specifying Normal Start Range (NSR) or Normal End Range (NER) values for workflow SLC milestones, remember that you must specify times using the 24-hour clock.

For example, if a Process is to start between 2:00 and 2:15 p.m., the time specified for NSRs and NSRe, respectively, must be 14:00 and 14:15. A typical mistake is to specify 02:00 and 02:15, but that means 2:00 to 2:15 in the morning.

Monitoring Windows

Another reason an SLC may not work as planned is that the events expected to trigger it occur outside its monitoring window.

In addition to the schedules associated with SLCs, SLC criteria include Start and End Window Tolerance values, which may range from zero to twenty-four hours.



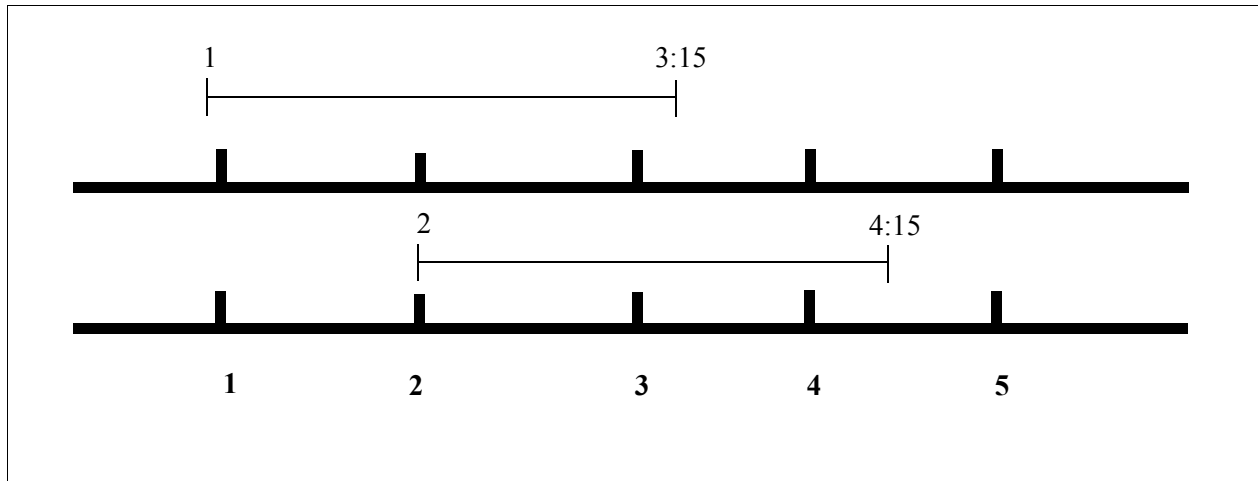
If the start and end events for the Process or file transfer being monitored occur outside the SLC's monitoring window, it will be as if the Process or transfer never occurred, and SLC events will be generated as such.

Another problem related to monitoring windows can occur when SLCs have multiple calendar schedules and those schedules overlap due to the size of the start or end window tolerance value specified.

Note: For SLCs and milestones with duration schedules, only the end window tolerance value is used.

Note: Sterling Control Center versions prior to 5.0 have a minimum window tolerance of one hour.

For example, if you have an SLC with two schedules, one that specifies an NERs of 02:00 and an NERe of 02:15, and a second that specifies an NERs of 03:00 and an NERe of 03:15, and the SLC has a start and end window tolerance of one hour, the monitoring windows for the two SLC instances will overlap and may cause unexpected events to be generated, as illustrated below:



The monitor window for the first SLC instance ranges from 1:00 to 3:15, while the monitor window for the second SLC instance ranges from 2:00 to 4:15, so if the first Process that should trigger the SLC does not run at all, but the second Process starts anytime between 2:00 and 3:15, rather than generating SLC events that indicate the first Process did not run but the second Process completed on time, the SLC events generated will indicate that the first Process completed late and the second Process did not run.

In cases such as this one, consider using a monitor tolerance value of zero hours and adjusting the SLC schedules as needed to avoid getting erroneous SLC events.

Monitored Servers

Back-level versions of monitored servers, that is, monitored servers not on current maintenance, may not generate statistics records, or may generate statistics records without the needed information, for the events constructed by Control Center to have the information needed to match the specified SLC parameters.

For example, when monitoring transfers of files to GDGs, the dataset name in the Process step start event must match the dataset name in the Process step end event, or SLCs do not work correctly. If the server being monitored is not on current maintenance, a known (and resolved) issue causes the dataset name in the Process step start event not to be fully resolved. That is, instead of the destination dataset name in the Process step start event being, for example, XYZZY.GDG.G0001V00, it is XYZZY.GDG(+1), while the dataset name in the Process step end event is fully resolved. If this occurs the SLC will not “work.” Even if the match criteria for the SLC is triggered by the dataset name in the step start event, because the dataset name in the step end event is different—the dataset name in the step end event is always the fully resolved GDG dataset name—the SLC logic does not recognize it as the terminating event.

You can tell if this issue is causing problems by closely examining the statistics records for the Process and seeing whether or not the dataset name in the step start and step end events match.

Note: For every schedule associated with an SLC, a separate SLC instance exists.

Note: The *Sterling Control Center Release Notes* is one resource to use to judge whether your monitored servers are at an appropriate maintenance level for monitoring by Control Center.

Rules and Actions

SLCs by themselves cause SLC events to be generated rather than alerts. You must have the appropriate rules and actions in place to watch for the SLC events and perform the actions you wish performed, including generating alerts and e-mails, when those SLCs are not met.

When shipped, Sterling Control Center comes with built-in rules that specify one of the built-in actions for all the various SLC events that may be generated. You must ensure that the appropriate rules and actions are in place, enabled or disabled, and in the appropriate priority order to get the results you want.

Note: Refer to *What Events May Be Generated By SLCs and When?* on page 97, and Appendix D, *Message IDs for Rules*, in the *Sterling Control Center System Administration Guide*, for the list of SLC type events that could occur.

Sterling Control Center Console - 127.0.0.1

Control Center Manage Monitor Configure Servers Tools Window Help

Servers Groups

- Control Center
 - ConnectDirect (3 / 3)
 - dander2k41
 - dander2k42
 - dander2k44
 - ConnectEnterprise (0 / 0)
 - Sterling Integrator (0 / 0)
 - File Transfer Protocol (0 / 0)

Rules - [Filter : None]

Filter: None (full listing shown)

Rules are applied in Priority order. Use the Move button to change the order.

#	✓	✎	⊞	⚡	Rule Name	Description
1	✓	✎	⊞	⚡	Process already running	CSLC031E - alert3
2	✓	✎	⊞	⚡	Process started early	CSLC032E - alert3
3	✓	✎	⊞	⚡	Process started on-time	CSLC033I - alert0
4	✓	✎	⊞	⚡	Process has not started	CSLC034E - alert2
5	✓	✎	⊞	⚡	Process started late	CSLC035E - alert2
6	✓	✎	⊞	⚡	Process ended early	CSLC036E - alert0
7	✓	✎	⊞	⚡	Process ended on-time	CSLC037I - alert0
8	✓	✎	⊞	⚡	Process has not ended	CSLC038E - alert2
9	✓	✎	⊞	⚡	Process ended late	CSLC039E - alert0
10	✓	✎	⊞	⚡	Process still not ended	CSLC040E - alert2
11	✓	✎	⊞	⚡	Process not started warn	CSLC047E - alert3
12	✓	✎	⊞	⚡	Process not ended warn	CSLC048E - alert3
13	✓	✎	⊞	⚡	Process not even started	CSLC026E - alert2
14	✓	✎	⊞	⚡	Process not started late	CSLC027E - alert1
15	✓	✎	⊞	⚡	Process suspended	CSLC046E - alert2
16	✓	✎	⊞	⚡	Proc duration short	CSLC041E - alert2
17	✓	✎	⊞	⚡	Proc duration okay	CSLC042I - alert0
18	✓	✎	⊞	⚡	Proc duration not ended	CSLC043E - alert2
19	✓	✎	⊞	⚡	Proc duration exceeded	CSLC044E - alert1
20	✓	✎	⊞	⚡	Proc duration way over	CSLC045E - alert1
21	✓	✎	⊞	⚡	Proc duration running	CSLC049E - alert3
22	✓	✎	⊞	⚡	Proc duration not determ.	CSLC050E - alert3
23	✓	✎	⊞	⚡	Proc duration suspended	CSLC046E - alert2
24	✓	✎	⊞	⚡	Trans running early	CSLC051E - alert3
25	✓	✎	⊞	⚡	Trans started early	CSLC052E - alert3
26	✓	✎	⊞	⚡	Trans started ontime	CSLC053I - alert0
27	✓	✎	⊞	⚡	Trans not started - late	CSLC054E - alert1
28	✓	✎	⊞	⚡	Trans started late	CSLC055E - alert2
29	✓	✎	⊞	⚡	Trans ended early	CSLC056E - alert3
30	✓	✎	⊞	⚡	Trans ended ontime	CSLC057I - alert0
31	✓	✎	⊞	⚡	Trans not ended - late	CSLC058E - alert2
32	✓	✎	⊞	⚡	Trans ended - late	CSLC059E - alert3
33	✓	✎	⊞	⚡	Trans still not ended	CSLC060E - alert2
34	✓	✎	⊞	⚡	Trans not started	CSLC067E - alert2
35	✓	✎	⊞	⚡	Trans not ended	CSLC068E - alert2
36	✓	✎	⊞	⚡	Trans not even started	CSLC028E - alert2
37	✓	✎	⊞	⚡	Trans not even start late	CSLC029E - alert2
38	✓	✎	⊞	⚡	Trans suspended	CSLC066E - alert2
39	✓	✎	⊞	⚡	Trans dur short	CSLC061E - alert2
40	✓	✎	⊞	⚡	Trans dur okay	CSLC062I - alert0
41	✓	✎	⊞	⚡	Trans dur exceeded	CSLC063E - alert2
42	✓	✎	⊞	⚡	Trans dur exceed-late	CSLC064E - alert2
43	✓	✎	⊞	⚡	Trans dur really late	CSLC065E - alert1
44	✓	✎	⊞	⚡	Trans dur running	CSLC069E - alert0
45	✓	✎	⊞	⚡	Trans dur not determined	CSLC070E - alert1
46	✓	✎	⊞	⚡	Trans dur suspended	CSLC066E - alert2
47	✓	✎	⊞	⚡	Transfer ahead of schedule - 25 dMin	CSLC171I - alert3
48	✓	✎	⊞	⚡	Transfer behind schedule - 25 dMin	CSLC172E - alert2
49	✓	✎	⊞	⚡	Transfer ahead of schedule - 50 dMin	CSLC173I - alert3
50	✓	✎	⊞	⚡	Transfer behind schedule - 50 dMin	CSLC174E - alert2
51	✓	✎	⊞	⚡	Transfer ahead of schedule - 75 dMin	CSLC175I - alert3
52	✓	✎	⊞	⚡	Transfer behind schedule - 75 dMin	CSLC176E - alert2
53	✓	✎	⊞	⚡	Transfer ahead of schedule - 25 dMax	CSLC177I - alert3
54	✓	✎	⊞	⚡	Transfer behind schedule - 25 dMax	CSLC178E - alert2
55	✓	✎	⊞	⚡	Transfer ahead of schedule - 50 dMax	CSLC179I - alert3
56	✓	✎	⊞	⚡	Transfer behind schedule - 50 dMax	CSLC180E - alert2
57	✓	✎	⊞	⚡	Transfer ahead of schedule - 75 dMax	CSLC181I - alert3

Find:

Move to Position #: 1 Move

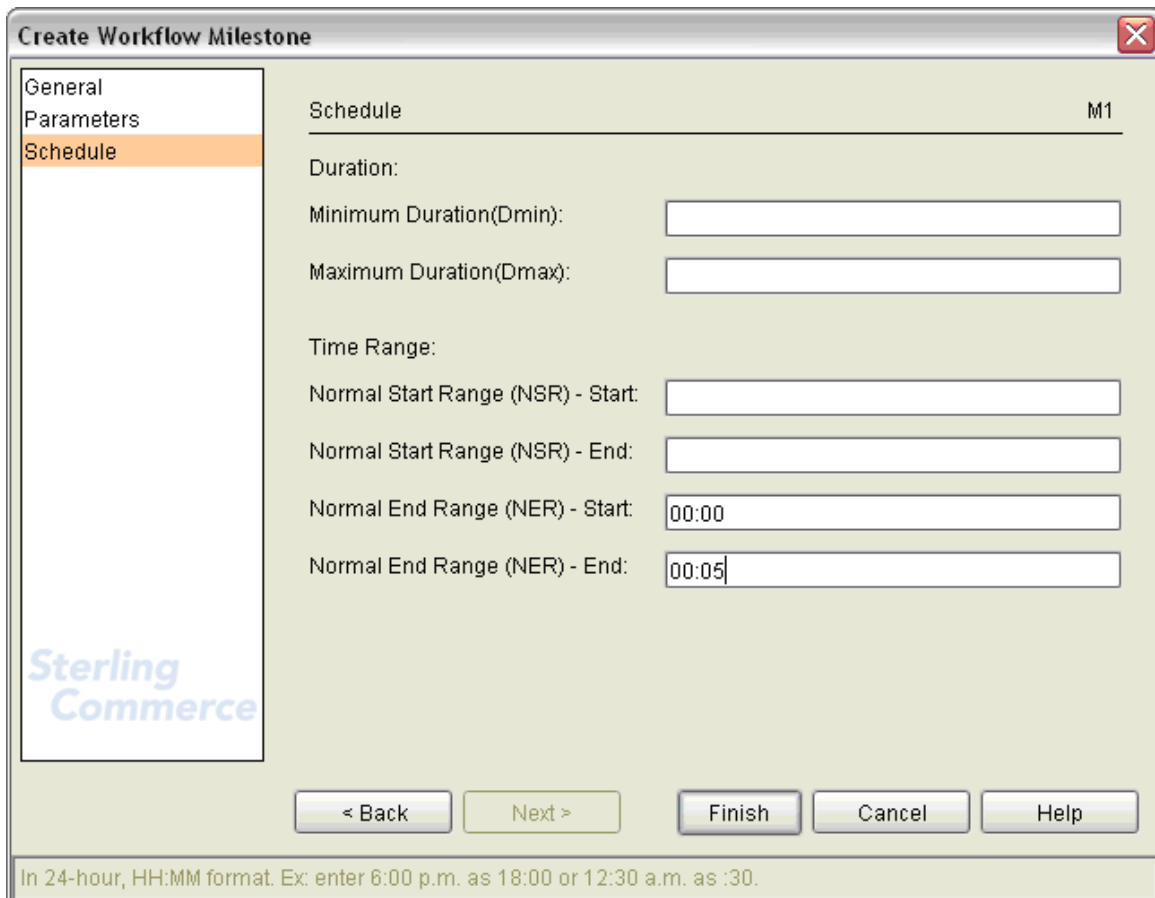
Memory Monitor Console: ██████████ Engine: ██████████

UTC: 14:17 Engine: 9:17 Local: 9:17 Preferred: 9:17 admin

Milestone Times

Values specified for Workflow SLC Milestone Normal Start Range and Normal End Range are offsets to the scheduled or actual start of the workflow being monitored, as opposed to clock times.

For example, if a workflow milestone should end no later than 1:05 a.m., you would not enter 01:05 as the Normal End Range end value, but rather the number of minutes relative to the schedule associated with the workflow which combined yield 1:05.



Create Workflow Milestone

General
Parameters
Schedule

Schedule M1

Duration:

Minimum Duration(Dmin):

Maximum Duration(Dmax):

Time Range:

Normal Start Range (NSR) - Start:

Normal Start Range (NSR) - End:

Normal End Range (NER) - Start:

Normal End Range (NER) - End:

Sterling Commerce

< Back Next > Finish Cancel Help

In 24-hour, HH:MM format. Ex: enter 6:00 p.m. as 18:00 or 12:30 a.m. as :30.

SLC Order

Unlike standard SLCs, wildcard and workflow SLCs must be ordered the way Control Center rules are. In the situation in which a single event matches the criteria of more than one wildcard SLC, or more than one workflow SLC's milestone, SLC order determines which SLC the event will trigger.

Time Zones

SLCs will not work properly during the change from daylight saving time to standard time, and vice versa, if the time zone parameter specified for Calendar Schedules associated with SLCs is not the same value as the time zone setting specified for the servers that you are monitoring.

Everything's Right Except the Alerts

Sometimes you may feel an SLC is not working because you are not seeing expected alerts in the Active Alerts Monitor.

Remember, alerts are nothing more than Sterling Control Center events with an additional element named Alert, along with an alert value (0, 1, 2, or 3). An alert element is added to an event by the rule service only if an enabled Control Center rule matches the event and specifies an action with an alert severity of 0, 1, 2, or 3. The Active Alerts Monitor displays all unhandled events that contain an alert element with a value greater than 0.

If you are not seeing expected alerts in the Active Alerts monitor, check to see if your situation resembles one of the following scenarios:

- Make sure that the rule that should be triggering the expected alerts has been properly defined by:
 - Specifying the appropriate SLC message ID
 - Associating an appropriate schedule and action with it
 - Enabling the rule
- The Active Alerts Monitor has the ability to overlay existing alerts with new correlated alerts. All SLC events contain an SLC instance identifier unique to each active SLC. The Active Alerts Monitor uses the SLC instance identifier to know when alerts it receives are correlated, and instead of displaying multiple alerts for the same SLC instance, displays only the latest.

For example, an SLC with a Normal Start Range (NSR) specified will generate an SLC notification with a message ID of CSLC047E if the NSR's time transpires without an appropriate starting event being seen, and another SLC notification with a message ID of CSLC034E if the NSR's time also transpires without an appropriate starting event being seen. In this case, the Active Alerts Monitor may first display the SLC event with the CSLC047E message, and then overlay that event with the SLC event containing the CSLC034E message, because their SLC_INSTANCE_ID values will be the same, which means they're correlated.

In fact, if the latest correlated event is an alert of severity 0, instead of displaying the newest alert in the view, the existing correlated alert (if one exists) is removed from the Active Alerts Monitor. For example, an SLC with a duration schedule generates an SLC notification with a message ID of CSLC049E because the dMin time passes without seeing an appropriate ending Process type event, and another SLC notification with a message ID of CSLC042I because an appropriate ending Process type event is seen before the dMax time occurs. If the rule triggered when an SLC notification with message ID CSLC042I occurs specifies an action with an alert severity of 0, then the alert with the message ID of CSLC049E is removed from the Active Alerts Monitor when the CSLC042I event occurs, instead of being replaced.

- The engine may not be collecting and processing events from monitored servers in a timely fashion. As a result, Control Center may issue contradictory messages. If you see out-of-compliance events for a Process or Process step, followed by an in-compliance event, this is typically caused by a lag between the time statistics are generated by monitored servers and the time statistics are collected and processed by Control Center.

For example, if an SLC-type event with a CSLC027E message (Process did not start by NERe) is followed by an SLC-type event with a CSLC037I message (Process ended within NERe) for

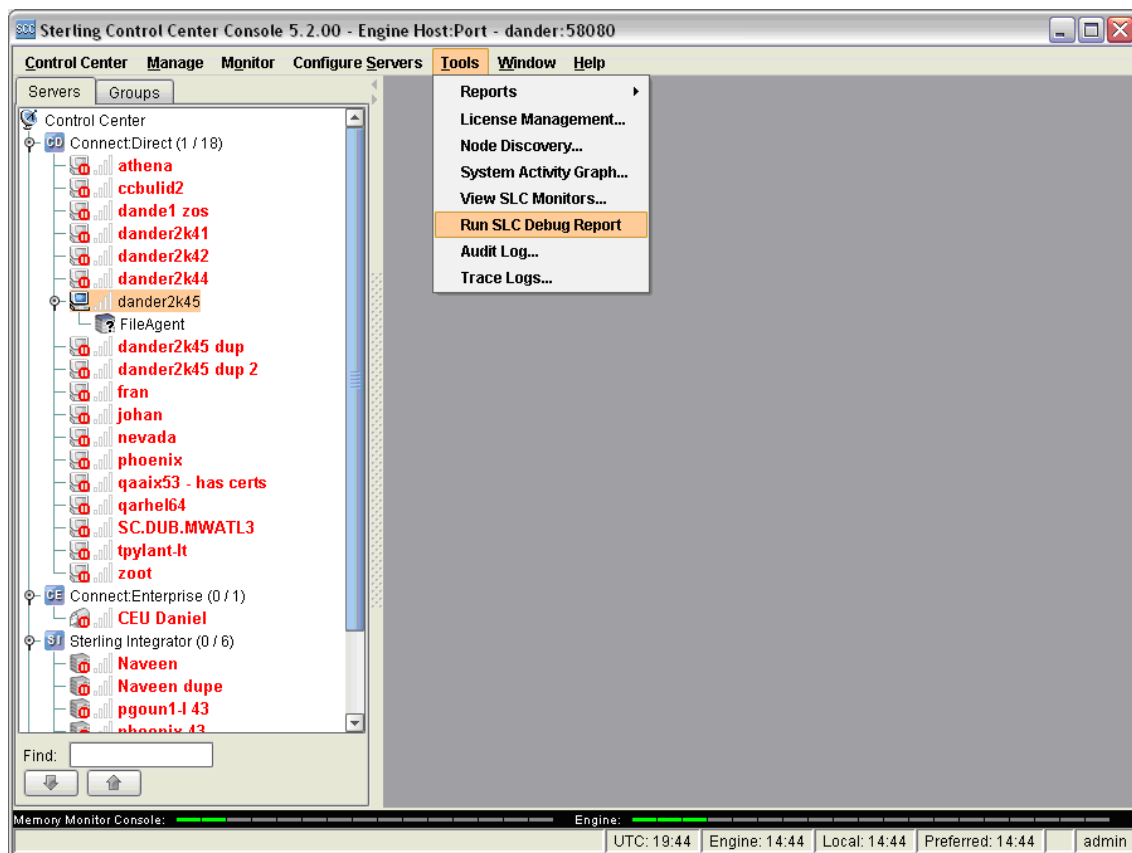
the same SLC instance, it is almost certainly caused by the engine not receiving and processing the Process end or Process step end statistic prior to the NERe even though the process ended within the NERe.

- The concurrence count for the SLC may not be set properly. For example, you may see more alerts than expected if the concurrence count is set too high for SLCs with calendar schedules, or you may see fewer alerts than expected if the concurrence count is set too low for SLCs with duration schedules.
- Make sure the alerts are associated with the server, servers, or server group for which you opened the Active Alerts monitor. Some SLC events, in particular the ones for events that did not occur, may not be associated with any servers and, in that case, the only way to see them is by opening an Active Alerts monitor for all of Control Center.

What Else Can I Do?

If you feel an SLC is not working but should be, check to see whether the expected SLC events were generated. One way to do this is to use the SLC Debug Report tool.

Select **Tools > Reports > Run SLC Debug Report**, fill in the dialog that appears with as much information as you can, and then click Run Report



At a minimum, specify the SLC ID for the SLC you are having trouble with and the start and end times for the specific period the problems occurred.

Server ID	Process Name	Process Number

At a minimum, you need to provide the specific SLC ID (the name of the SLC Group) you want the report to focus on and a time frame for which the tool will look for events related to the SLC specified. You have the option to specify process information. If you choose to do so, you must specify all of the following pieces of information: Sterling Control Center Server ID, the Process Name, and Process number for each Process specified.

Note: Note the format shown for the Start Time and End Time (YYYY/MM/DD HH:MM) must be used exactly if the default values are altered.

The resulting report will appear in your default browser, in either a new tab, or a new window depending on how your browser has been configured.

You may subsequently print this information, or copy and paste it into an e-mail to send to someone to get help with your SLC.

The report will include:

- The definition of the SLC
- The definition of all schedules referenced by the SLC
- All related SLC events produced for the SLC specified in the time frame given, including any events that triggered the SLC events, such as a Process Start event
- The definition of any rules that were triggered by the SLC events generated

- The definition of any actions that were taken by the rules triggered
- The engine's e-mail system settings

If any process information is specified, the resulting report will also include all the process-related events for those processes.

The following shows a sample SLC Debug Report:

```

SLC Debug Utility Version 1.0
=====
Date/Time: Wed Aug 04 15:13:09 CDT 2010
SLC ID: bothStandardSLC1
Start Range: Wed Aug 04 00:00:00 CDT 2010
End Range: Wed Aug 04 17:00:00 CDT 2010
Process ID: [Name=sample2 #28754 Server= aix534000sp]

bothStandardSLC1:
<?xml version="1.0" encoding="UTF-8"?>
<slaGroup>
  <id>bothStandardSLC1</id>
  <name>bothStandardSLC1</name>
  <desc></desc>
  <enabled>>false</enabled>
  <nodes>
    <node>aix534000sp</node>
  </nodes>
  <processes>
    <process>sample2</process>
  </processes>
  <fileNames></fileNames>
  <submitters></submitters>
  <remoteNodes></remoteNodes>
  <schedules>
    <schedule>bothDuration1</schedule>
  </schedules>
  <serverGroups></serverGroups>
  <visibilityGroups>
    <visibilityGroup>Both</visibilityGroup>
  </visibilityGroups>
  <missingEvent>>true</missingEvent>
  <monitorToleranceStart>1</monitorToleranceStart>
  <monitorToleranceEnd>1</monitorToleranceEnd>
</slaGroup>

Schedules:

bothDuration1:
<?xml version="1.0" encoding="UTF-8"?>
<durationSchedule>
  <id>bothDuration1</id>
  <desc></desc>
  <enabled>>true</enabled>
  <simpleSlc>>false</simpleSlc>
  <dMin>000:00:00</dMin>
  <dMax>000:02:00</dMax>
  <vistype>private</vistype>
  <roles>
    <role>bothRole</role>
  </roles>
</durationSchedule>

```

SLC Event Report:

Event # 1

Event ID: 83947193024450792

Event Time: 2010/08/04 19:27:46.870 UTC

Processed Time: 2010/08/04 19:27:46.886|2010/08/04 19:27:46.883 UTC

Event Type: SLC

Message: CSLC033I Process started. SLC:

bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1.83947193024450767

SLC Instance ID: 83947193024450767

Rule Triggered: Process started on-time|Process started on-time-both

Alert: 0|3

Triggering event:

```
<event><alert>|</alert><ruleInstanceId>|</ruleInstanceId><actionId>|</actionId><rule
Id>|</ruleId><localNode>P</localNode><nodeName>aix534000sp</nodeName><dvg>Both</dvg>
<actionsFlag>2010/08/04 19:27:46.945|2010/08/04
19:27:46.851</actionsFlag><dateTime>2010/08/04
19:27:49.000</dateTime><emailFlag>0|0</emailFlag><eventId>83947193024450786</eventId
><eventType>3</eventType><fileSize>-1</fileSize><messageId>XSMG200I</messageId><node
Id>aix534000sp</nodeId><nodeType>1</nodeType><origNode>aix534000sp</origNode><percen
tComplete>-1</percentComplete><processId>28754</processId><processName>sample2</proc
essName><remoteNode>hpig4000sp</remoteNode><returnCode>0</returnCode><seqNum>1</seqN
um><shortText>Process started, process:28754 name:sample2
SNode:hpig4000sp</shortText><slaFlag>0</slaFlag><submitterId>qatest</submitterId><tr
apFlag>0|0</trapFlag><userOpFlag>0|0</userOpFlag><SUBI>qatest@aix534000sp</SUBI><SBN
D>aix534000sp</SBND><STPT>2010/08/04 19:27:49.000</STPT><STRT>2010/08/04
19:27:49.000</STRT><PRTY>10</PRTY><CLAS>1</CLAS><recCat>CAPR</recCat><recordId>PSTR<
/recordId></event>
```

Event XML:

```
<event><ruleInstanceId>|</ruleInstanceId><activityType>P</activityType><sourceEventT
ime>1280950069000</sourceEventTime><groupId>bothStandardSLC1</groupId><dvg>Both</dvg
><actionsFlag>2010/08/04 19:27:46.886|2010/08/04
19:27:46.883</actionsFlag><actionId>[alert0]|[alert3]</actionId><alert>0|3</alert><d
ateTime>2010/08/04
19:27:46.870</dateTime><emailFlag>0|0</emailFlag><eventId>83947193024450792</eventId
><eventType>6</eventType><fileSize>-1</fileSize><messageId>CSLC033I</messageId><node
Id>aix534000sp</nodeId><nodeType>1</nodeType><percentComplete>-1</percentComplete><p
rocessId>28754</processId><processName>sample2</processName><remoteNode>hpig4000sp</
remoteNode><returnCode>0</returnCode><ruleId>Process started on-time|Process started
on-time-both</ruleId><seqNum>83947193024450792</seqNum><shortText>CSLC033I Process
started. SLC:
bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1.83947193024450767</shortTe
xt><slaFlag>0</slaFlag><slcId>bothStandardSLC1.bothDuration1.aix534000sp.sample2.tru
e.1.83947193024450767</slcId><slcInstanceId>83947193024450767</slcInstanceId><slcSou
rce1>bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1</slcSource1><slcSourc
e2>aix534000sp.sample2.28754.aix534000sp</slcSource2><slcSourceEventId>8394719302445
0786</slcSourceEventId><trapFlag>0|0</trapFlag><userOpFlag>0|0</userOpFlag></event>
```

```

Event # 2
Event ID: 83947193024450794
Event Time: 2010/08/04 19:27:47.438 UTC
Processed Time: 2010/08/04 19:27:47.448|2010/08/04 19:27:47.445 UTC
Event Type: SLC
Message: CSLC042I Process ended when expected - between dMin and dMax. SLC:
bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1.83947193024450767
SLC Instance ID: 83947193024450767
Rule Triggered: Proc duration okay|Proc duration okay - both
Alert: 0|3
Triggering event:
<event><alert>|</alert><ruleInstanceId>|</ruleInstanceId><actionId>|</actionId><ruleId>|</ruleId><localNode>P</localNode><nodeName>aix534000sp</nodeName><dvgt>Both</dvgt><actionsFlag>2010/08/04 19:27:47.465|2010/08/04 19:27:47.432</actionsFlag><dateTime>2010/08/04 19:27:50.000</dateTime><emailFlag>0|0</emailFlag><eventId>83947193024450791</eventId><eventType>4</eventType><fileSize>-1</fileSize><messageId>XSMG252I</messageId><nodeId>aix534000sp</nodeId><nodeType>1</nodeType><origNode>aix534000sp</origNode><percentComplete>-1</percentComplete><processId>28754</processId><processName>sample2</processName><remoteNode>hpig4000sp</remoteNode><returnCode>0</returnCode><seqNum>1</seqNum><shortText>Process completed,
SNODE:hpig4000sp</shortText><slaFlag>0</slaFlag><submitterId>qatest</submitterId><trapFlag>0|0</trapFlag><userOpFlag>0|0</userOpFlag><SUBI>qatest@aix534000sp</SUBI><SBND>aix534000sp</SBND><STPT>2010/08/04 19:27:50.000</STPT><STRT>2010/08/04 19:27:49.000</STRT><recCat>CAPR</recCat><recordId>PRED</recordId></event>
Event XML:
<event><ruleInstanceId>|</ruleInstanceId><activityType>P</activityType><sourceEventTime>1280950070000</sourceEventTime><groupId>bothStandardSLC1</groupId><dvgt>Both</dvgt><actionsFlag>2010/08/04 19:27:47.448|2010/08/04 19:27:47.445</actionsFlag><actionId>[alert0]|[alert3]</actionId><alert>0|3</alert><dateTime>2010/08/04 19:27:47.438</dateTime><emailFlag>0|0</emailFlag><eventId>83947193024450794</eventId><eventType>6</eventType><fileSize>-1</fileSize><messageId>CSLC042I</messageId><nodeId>aix534000sp</nodeId><nodeType>1</nodeType><percentComplete>-1</percentComplete><processId>28754</processId><processName>sample2</processName><remoteNode>hpig4000sp</remoteNode><returnCode>0</returnCode><ruleId>Proc duration okay|Proc duration okay - both</ruleId><seqNum>83947193024450794</seqNum><shortText>CSLC042I Process ended when expected - between dMin and dMax. SLC:
bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1.83947193024450767</shortText><slaFlag>0</slaFlag><slcId>bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1.83947193024450767</slcId><slcInstanceId>83947193024450767</slcInstanceId><slcSource1>bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1</slcSource1><slcSource2>aix534000sp.sample2.28754.aix534000sp</slcSource2><slcSourceEventId>83947193024450791</slcSourceEventId><trapFlag>0|0</trapFlag><userOpFlag>0|0</userOpFlag></event>

Rules:
Process started on-time-both:
<?xml version="1.0" encoding="UTF-8"?>
<rule>
  <actionId>alert3</actionId>
  <desc>CSLC033I - alert3</desc>
  <enabled>true</enabled>
  <eventTypes>
    <eventType>6</eventType>
  </eventTypes>
  <id>Process started on-time-both</id>

```

```

<match>&quot;/event[contains(messageId, &apos;CSLC033I&apos;)]&quot;</match>
<priority>1</priority>
<relative_priority>1</relative_priority>
<visibilityGroups>
  <visibilityGroup>Both</visibilityGroup>
</visibilityGroups>
<linkedRule>
  <enabled>>false</enabled>
  <nonResolutionActionId>alert0</nonResolutionActionId>
  <resolutionActionId>alert0</resolutionActionId>
  <resolutionTimeout>5</resolutionTimeout>
</linkedRule>
<effectiveMatch>&quot;/event[contains(messageId, &apos;CSLC033I&apos;)] and (
eventType = &apos;6&apos;)]&quot;</effectiveMatch></rule>

```

Proc duration okay:

```

<?xml version="1.0" encoding="UTF-8"?>
<rule>
  <actionId>alert0</actionId>
  <desc>CSLC042I - alert0</desc>
  <enabled>true</enabled>
  <id>Proc duration okay</id>
  <match>&quot;/event[eventType=&apos;6&apos;] and
contains(messageId, &apos;CSLC042I&apos;)]&quot;</match>

  <priority>21</priority><eventType>6</eventType></rule>

```

Process started on-time:

```

<?xml version="1.0" encoding="UTF-8"?>
<rule>
  <actionId>alert0</actionId>
  <desc>CSLC033I - alert0</desc>
  <enabled>true</enabled>
  <id>Process started on-time</id>
  <match>&quot;/event[eventType=&apos;6&apos;] and
contains(messageId, &apos;CSLC033I&apos;)]&quot;</match>

  <priority>7</priority><eventType>6</eventType></rule>

```

Proc duration okay - both:

```

<?xml version="1.0" encoding="UTF-8"?>
<rule>
  <actionId>alert3</actionId>
  <desc>CSLC042I - alert3</desc>
  <enabled>true</enabled>
  <eventTypes>
    <eventType>6</eventType>
  </eventTypes>
  <id>Proc duration okay - both</id>
  <match>&quot;/event[contains(messageId, &apos;CSLC042I&apos;)]&quot;</match>
  <priority>8</priority>
  <relative_priority>8</relative_priority>
  <visibilityGroups>
    <visibilityGroup>Both</visibilityGroup>
  </visibilityGroups>

```



```

<linkedRule>
  <enabled>>false</enabled>
  <nonResolutionActionId>alert0</nonResolutionActionId>
  <resolutionActionId>alert0</resolutionActionId>
  <resolutionTimeout>5</resolutionTimeout>
</linkedRule>
<effectiveMatch>&quot;/event [contains (messageId, &apos;CSLC042I&apos; ) and (
eventType = &apos;6&apos; ) ]&quot;</effectiveMatch></rule>

```

Actions:

```

alert0:
<?xml version="1.0" encoding="UTF-8"?>
<action>
  <alert>0</alert>
  <desc></desc>
  <id>alert0</id>
</action>

```

```

alert3:
<?xml version="1.0" encoding="UTF-8"?>
<action>
  <alert>3</alert>
  <desc></desc>
  <id>alert3</id>
</action>

```

Engine Email System Settings:

```

SMTP Host Name: shuttlerun
SMTP Host Port: 25

```

Events Related to Process ID: [Name=sample2 #28754 Server= aix534000sp]

```

Event # 1
Event ID: 83947193024450783
Event Time: 2010/08/04 19:27:48.000 UTC
Processed Time: 2010/08/04 19:27:46.804|2010/08/04 19:27:46.775 UTC
Event Type: Process status
Record ID: QCEX
Message: CNCD053I Process moved to Execution queue.
Rule Triggered: |
Alert: |
Event XML:
<event><alert>|</alert><ruleInstanceId>|</ruleInstanceId><actionId>|</actionId><ruleId>|</ruleId><processQueue>EXEC</processQueue><nodeName>aix534000sp</nodeName><localNode>P</localNode><dvg>Both</dvg><actionsFlag>2010/08/04 19:27:46.804|2010/08/04 19:27:46.775</actionsFlag><dateTime>2010/08/04 19:27:48.000</dateTime><eMailFlag>0|0</eMailFlag><eventId>83947193024450783</eventId><eventType>9</eventType><fileSize>-1</fileSize><messageId>CNCD053I</messageId><nodeId>aix534000sp</nodeId><nodeType>1</nodeType><origNode>aix534000sp</origNode><percentComplete>-

```

```
1</percentComplete><processId>28754</processId><processName>sample2</processName><remoteNode>hpig4000sp</remoteNode><seqNum>0</seqNum><shortText>CNCD053I Process moved to Execution queue.</shortText><slaFlag>0</slaFlag><submitterId>qatest</submitterId><trapFlag>0|0</trapFlag><userOpFlag>0|0</userOpFlag><SUBI>qatest@aix534000sp</SUBI><SBND>aix534000sp</SBND><status>PE</status><recCat>CAEV</recCat><recordId>QCEX</recordId></event>
```

Event # 2

Event ID: 83947193024450786

Event Time: 2010/08/04 19:27:49.000 UTC

Processed Time: 2010/08/04 19:27:46.945|2010/08/04 19:27:46.851 UTC

Event Type: Process started

Record ID: PSTR

Message: Process started, process:28754 name:sample2 SNODE:hpig4000sp

Rule Triggered: |

Alert: |

Event XML:

```
<event><alert>|</alert><ruleInstanceId>|</ruleInstanceId><actionId>|</actionId><ruleId>|</ruleId><localNode>P</localNode><nodeName>aix534000sp</nodeName><divg>Both</divg><actionsFlag>2010/08/04 19:27:46.945|2010/08/04 19:27:46.851</actionsFlag><dateTime>2010/08/04 19:27:49.000</dateTime><emailFlag>0|0</emailFlag><eventId>83947193024450786</eventId><eventType>3</eventType><fileSize>-1</fileSize><messageId>XSMG200I</messageId><nodeId>aix534000sp</nodeId><nodeType>1</nodeType><origNode>aix534000sp</origNode><percentComplete>-1</percentComplete><processId>28754</processId><processName>sample2</processName><remoteNode>hpig4000sp</remoteNode><returnCode>0</returnCode><seqNum>1</seqNum><shortText>Process started, process:28754 name:sample2 SNODE:hpig4000sp</shortText><slaFlag>0</slaFlag><submitterId>qatest</submitterId><trapFlag>0|0</trapFlag><userOpFlag>0|0</userOpFlag><SUBI>qatest@aix534000sp</SUBI><SBND>aix534000sp</SBND><STPT>2010/08/04 19:27:49.000</STPT><STRT>2010/08/04 19:27:49.000</STRT><PRTY>10</PRTY><CLAS>1</CLAS><recCat>CAPR</recCat><recordId>PSTR</recordId></event>
```

Event # 3

Event ID: 83947193024450787

Event Time: 2010/08/04 19:27:49.000 UTC

Processed Time: 2010/08/04 19:27:46.962|2010/08/04 19:27:46.948|2010/08/04 19:27:46.958 UTC

Event Type: Process step started

Record ID: LSST

Message: Local Step started.

Rule Triggered: ||

Alert: ||

Event XML:

```
<event><alert>||</alert><ruleInstanceId>||</ruleInstanceId><actionId>||</actionId><ruleId>||</ruleId><localNode>P</localNode><nodeName>aix534000sp</nodeName><direction>outBound</direction><divg>Both|Up</divg><actionsFlag>2010/08/04 19:27:46.962|2010/08/04 19:27:46.948|2010/08/04 19:27:46.958</actionsFlag><dateTime>2010/08/04 19:27:49.000</dateTime><destFile>/data/cd4000sp/ndm/cfg/upfile.from_aix</destFile><emailFlag>0|0</emailFlag><eventId>83947193024450787</eventId><eventType>1</eventType><fileSize>-1</fileSize><fromNode>P</fromNode><messageId>XSMG201I</messageId><nodeId>aix534000sp</nodeId><nodeType>1</nodeType><origNode>aix534000sp</origNode><percentComplete>-1</percentComplete><processId>28754</processId><processName>sample2</processName><remoteNode>hpig4000sp</remoteNode><seqNum>2</seqNum><shortText>Local Step
```

```

cfg/msgfile.cfg</sourceFile><stepName>Up</stepName><submitterId>qatest</submitterId><
trapFlag>0|0|0</trapFlag><userOpFlag>0|0|0</userOpFlag><SUBI>qatest@aix534000sp</SUBI
><SBND>aix534000sp</SBND><STRT>2010/08/04
19:27:49.000</STRT><RSTR>N</RSTR><recCat>CAPR</recCat><recordId>LSST</recordId></even
t>

```

Event # 4

Event ID: 83947193024450788

Event Time: 2010/08/04 19:27:49.000 UTC

Processed Time: 2010/08/04 19:27:47.331|2010/08/04 19:27:47.279|2010/08/04

19:27:47.326 UTC

Event Type: Process step ended

Record ID: CTRC

Message: Copy step successful.

Rule Triggered: |bothRule|upRule

Alert: |1|1

Event XML:

```

<event><alert>|1|1</alert><ruleInstanceId>||</ruleInstanceId><actionId>|[bothAlert1]|
[upAlert1]</actionId><ruleId>|bothRule|upRule</ruleId><localNode>P</localNode><nodeNa
me>aix534000sp</nodeName><direction>outBound</direction><dvg>Both|Up</dvg><actionsFla
g>2010/08/04 19:27:47.331|2010/08/04 19:27:47.279|2010/08/04
19:27:47.326</actionsFlag><dateTime>2010/08/04
19:27:49.000</dateTime><destFile>/data/cd4000sp/ndm/cfg/upfile.from_aix</destFile><em
ailFlag>0|1280950067279|1280950067326</emailFlag><eventId>83947193024450788</eventId>
<eventType>2</eventType><fileSize>4313802</fileSize><fromNode>P</fromNode><messageId>
SCPA000I</messageId><nodeId>aix534000sp</nodeId><nodeType>1</nodeType><origNode>aix53
4000sp</origNode><percentComplete>-1</percentComplete><processId>28754</processId><pr
ocessName>sample2</processName><remoteNode>hpig4000sp</remoteNode><returnCode>0</retu
rnCode><seqNum>3</seqNum><shortText>Copy step
successful.</shortText><slaFlag>0</slaFlag><sourceFile>/data/cd4000sp/ndm/cfg/msgfile
.cfg</sourceFile><stepName>Up</stepName><submitterId>qatest</submitterId><trapFlag>0|
0|0</trapFlag><userOpFlag>0|0|0</userOpFlag><SUBI>qatest@aix534000sp</SUBI><SBND>aix5
34000sp</SBND><STPT>2010/08/04 19:27:49.000</STPT><STRT>2010/08/04
19:27:49.000</STRT><certIssuer>(O=SCI/L=Irving/ST=Texas/C=US/SHA1=fed0aa6547066dccfc
7476c19428b3a000b9c1e3/SN=00)</certIssuer><certSubject>(C=US/ST=Texas/O=SCI/OU=SV/CN
=cdwopxp01.csg.stercomm.com/emailAddress=jason_legel@stercomm.com/SHA1=e40edf02b0bb9
3463fb213f364600f7ae5551ad7/SN=7B)</certSubject><secureEnabled>Y</secureEnabled><secu
reProtocol>TLSv1</secureProtocol><cipherSuite>TLS_DHE_RSA_WITH_AES_256_CBC_SHA</ciphe
rSuite><OMSG>SCPA000I</OMSG><OCCD>0</OCCD><LMSG>SCPA000I</LMSG><LCCD>0</LCCD><RUSZ>16
000</RUSZ><RSTR>N</RSTR><XLAT>N</XLAT><LKFL>N</LKFL><recCat>CAPR</recCat><CKPT>Y</CKP
T><ECMP>Y</ECMP><SCMP>N</SCMP><DDS3>R</DDS3><bytesSent>4313802</bytesSent><bytesRead>
4313802</bytesRead><SRCR>188</SRCR><SRUX>58</SRUX><DBYW>4313802</DBYW><DRCW>188</DRCW
><DBYX>885370</DBYX><DRUX>58</DRUX><recordId>CTRC</recordId></event>

```

Event # 5

Event ID: 83947193024450789

Event Time: 2010/08/04 19:27:49.000 UTC

Processed Time: 2010/08/04 19:27:47.351|2010/08/04 19:27:47.335|2010/08/04

19:27:47.347 UTC

Event Type: Process step started

Record ID: LSST

Message: Local Step started.

Rule Triggered: ||

Alert: ||

Event XML:
<event><alert>||</alert><ruleInstanceId>||</ruleInstanceId><actionId>||</actionId><ruleId>||</ruleId><localNode>P</localNode><nodeName>aix534000sp</nodeName><direction>in
Bound</direction><dvg>Both|Down</dvg><actionsFlag>2010/08/04 19:27:47.351|2010/08/04
19:27:47.335|2010/08/04 19:27:47.347</actionsFlag><dateTime>2010/08/04
19:27:49.000</dateTime><destFile>/data/cd4000sp/ndm/cfg/downfile.from_hp</destFile><e
MailFlag>0|0|0</eMailFlag><eventId>83947193024450789</eventId><eventType>1</eventType
><fileSize>-1</fileSize><fromNode>S</fromNode><messageId>XSMG201I</messageId><nodeId>
aix534000sp</nodeId><nodeType>1</nodeType><origNode>aix534000sp</origNode><percentCom
plete>-1</percentComplete><processId>28754</processId><processName>sample2</processNa
me><remoteNode>hpig4000sp</remoteNode><seqNum>4</seqNum><shortText>Local Step
started.</shortText><slaFlag>0</slaFlag><sourceFile>/data/cd4000sp/ndm/cfg/upfile.fro
m_aix</sourceFile><stepName>Down</stepName><submitterId>qatest</submitterId><trapFlag
>0|0|0</trapFlag><userOpFlag>0|0|0</userOpFlag><SUBI>qatest@aix534000sp</SUBI><SBND>a
ix534000sp</SBND><STRT>2010/08/04
19:27:49.000</STRT><RSTR>N</RSTR><recCat>CAPR</recCat><recordId>LSST</recordId></even
t>

Event # 6

Event ID: 83947193024450790

Event Time: 2010/08/04 19:27:50.000 UTC

Processed Time: 2010/08/04 19:27:47.429|2010/08/04 19:27:47.383|2010/08/04

19:27:47.424 UTC

Event Type: Process step ended

Record ID: CTRC

Message: Copy step successful.

Rule Triggered: |bothRule|downRule

Alert: |1|1

Event XML:

<event><alert>|1|1</alert><ruleInstanceId>||</ruleInstanceId><actionId>|[bothAlert1] |
[downAlert1]</actionId><ruleId>|bothRule|downRule</ruleId><localNode>P</localNode><no
deName>aix534000sp</nodeName><direction>inB
ound</direction><dvg>Both|Down</dvg><actionsFlag>2010/08/04 19:27:47.429|2010/08/04
19:27:47.383|2010/08/04 19:27:47.424</actionsFlag><dateTime>2010/08/04
19:27:50.000</dateTime><destFile>/data/cd4000sp/ndm/cfg/downfile.from_hp</destFile><e
MailFlag>0|1280950067383|1280950067424</eMailFlag><eventId>83947193024450790</eventId
><eventType>2</eventType><fileSize>4313802</fileSize><fromNode>S</fromNode><messageId
>SCPA000I</messageId><nodeId>aix534000sp</nodeId><nodeType>1</nodeType><origNode>aix5
34000sp</origNode><percentComplete>-1</percentComplete><processId>28754</processId><p
rocessName>sample2</processName><remoteNode>hpig4000sp</remoteNode><returnCode>0</ret
urnCode><seqNum>0</seqNum><shortText>Copy step
successful.</shortText><slaFlag>0</slaFlag><sourceFile>/data/cd4000sp/ndm/cfg/upfile.
from_aix</sourceFile><stepName>Down</stepName><submitterId>qatest</submitterId><trapF
lag>0|0|0</trapFlag><userOpFlag>0|0|0</userOpFlag><SUBI>qatest@aix534000sp</SUBI><SBN
D>aix534000sp</SBND><STPT>2010/08/04 19:27:50.000</STPT><STRT>2010/08/04
19:27:49.000</STRT><certIssuer>(/O=SCI/L=Irving/ST=Texas/C=US/SHA1=fed0aa6547066dccfc
7476c19428b3a000b9c1e3/SN=00)</certIssuer><certSubject>(/C=US/ST=Texas/O=SCI/OU=SV/CN
=cdwospxp01.csg.stercomm.com/emailAddress=jason_legel@stercomm.com/SHA1=e40edf02b0bb9
3463fb213f364600f7ae5551ad7/SN=7B)</certSubject><secureEnabled>Y</secureEnabled><secu
reProtocol>TLSv1</secureProtocol><cipherSuite>TLS_DHE_RSA_WITH_AES_256_CBC_SHA</ciphe
rSuite><OMSG>SCPA000I</OMSG><OCCD>0</OCCD><LMSG>SCPA000I</LMSG><LCCD>0</LCCD><RUSZ>16
000</RUSZ><RSTR>N</RSTR><XLAT>N</XLAT><LKFL>N</LKFL><recCat>CAPR</recCat><CKPT>Y</CKP
T><ECMP>Y</ECMP><SCMP>N</SCMP><DDS3>R</DDS3><bytesSent>4313802</bytesSent><bytesRead>
4313802</bytesRead><SRCR>188</SRCR><SRUX>58</SRUX><DBYW>4313802</DBYW><DRCW>188</DRCW
><DBYX>885370</DBYX><DRUX>58</DRUX><recordId>CTRC</recordId></event>

```

Event # 7
Event ID: 83947193024450791
Event Time: 2010/08/04 19:27:50.000 UTC
Processed Time: 2010/08/04 19:27:47.465|2010/08/04 19:27:47.432 UTC
Event Type: Process ended
Record ID: PRED
Message: Process completed, SNODE:hpig4000sp
Rule Triggered: |
Alert: |
Event XML:
<event><alert>|</alert><ruleInstanceId>|</ruleInstanceId><actionId>|</actionId><rule
Id>|</ruleId><localNode>P</localNode><nodeName>aix534000sp</nodeName><dvg>Both</dvg>
<actionsFlag>2010/08/04 19:27:47.465|2010/08/04
19:27:47.432</actionsFlag><dateTime>2010/08/04
19:27:50.000</dateTime><emailFlag>0|0</emailFlag><eventId>83947193024450791</eventId
><eventType>4</eventType><fileSize>-1</fileSize><messageId>XSMG252I</messageId><node
Id>aix534000sp</nodeId><nodeType>1</nodeType><origNode>aix534000sp</origNode><percen
tComplete>-1</percentComplete><processId>28754</processId><processName>sample2</proc
essName><remoteNode>hpig4000sp</remoteNode><returnCode>0</returnCode><seqNum>1</seqN
um><shortText>Process completed,
SNODE:hpig4000sp</shortText><slaFlag>0</slaFlag><submitterId>qatest</submitterId><tr
apFlag>0|0</trapFlag><userOpFlag>0|0</userOpFlag><SUBI>qatest@aix534000sp</SUBI><SBN
D>aix534000sp</SBND><STPT>2010/08/04 19:27:50.000</STPT><STRT>2010/08/04
19:27:49.000</STRT><recCat>CAPR</recCat><recordId>PRED</recordId></event>

Event # 8
Event ID: 83947193024450792
Event Time: 2010/08/04 19:27:46.870 UTC
Processed Time: 2010/08/04 19:27:46.886|2010/08/04 19:27:46.883 UTC
Event Type: SLC
Message: CSLC033I Process started. SLC:
bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1.83947193024450767
SLC Instance ID: 83947193024450767
Rule Triggered: Process started on-time|Process started on-time-both
Alert: 0|3
Triggering event:
<event><alert>|</alert><ruleInstanceId>|</ruleInstanceId><actionId>|</actionId><rule
Id>|</ruleId><localNode>P</localNode><nodeName>aix534000sp</nodeName><dvg>Both</dvg>
<actionsFlag>2010/08/04 19:27:46.945|2010/08/04
19:27:46.851</actionsFlag><dateTime>2010/08/04
19:27:49.000</dateTime><emailFlag>0|0</emailFlag><eventId>83947193024450786</eventId
><eventType>3</eventType><fileSize>-1</fileSize><messageId>XSMG200I</messageId><node
Id>aix534000sp</nodeId><nodeType>1</nodeType><origNode>aix534000sp</origNode><percen
tComplete>-
1</percentComplete><processId>28754</processId><processName>sample2</processName><re
moteNode>hpig4000sp</remoteNode><returnCode>0</returnCode><seqNum>1</seqNum><shortTe
xt>Process started, process:28754 name:sample2
SNODE:hpig4000sp</shortText><slaFlag>0</slaFlag><submitterId>qatest</submitterId><tr
apFlag>0|0</trapFlag><userOpFlag>0|0</userOpFlag><SUBI>qatest@aix534000sp</SUBI><SBN
D>aix534000sp</SBND><STPT>2010/08/04 19:27:49.000</STPT><STRT>2010/08/04
19:27:49.000</STRT><PRTY>10</PRTY><CLAS>1</CLAS><recCat>CAPR</recCat><recordId>PSTR<
/recordId></event>
Event XML:
<event><ruleInstanceId>|</ruleInstanceId><activityType>P</activityType><sourceEventT
ime>1280950069000</sourceEventTime><groupId>bothStandardSLC1</groupId><dvg>Both</dvg
><actionsFlag>2010/08/04 19:27:46.886|2010/08/04
19:27:46.883</actionsFlag><actionId>[alert0]| [alert3]</actionId><alert>0|3</alert><d
ateTime>2010/08/04

```

```
19:27:46.870</dateTime><emailFlag>0|0</emailFlag><eventId>83947193024450792</eventId>
<eventType>6</eventType><fileSize>-1</fileSize><messageId>CSLC033I</messageId><nodeId
>aix534000sp</nodeId><nodeType>1</nodeType><percentComplete>-1</percentComplete><proc
essId>28754</processId><processName>sample2</processName><remoteNode>hpig4000sp</remo
teNode><returnCode>0</returnCode><ruleId>Process started on-time|Process started
on-time-both</ruleId><seqNum>83947193024450792</seqNum><shortText>CSLC033I Process
started. SLC:
bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1.83947193024450767</shortTex
t><slaFlag>0</slaFlag><slcId>bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.
1.83947193024450767</slcId><slcInstanceId>83947193024450767</slcInstanceId><slcSource
1>bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1</slcSource1><slcSource2>a
ix534000sp.sample2.28754.aix534000sp</slcSource2><slcSourceEventId>83947193024450786<
/slcSourceEventId><trapFlag>0|0</trapFlag><userOpFlag>0|0</userOpFlag></event>
```

Event # 9

Event ID: 83947193024450794

Event Time: 2010/08/04 19:27:47.438 UTC

Processed Time: 2010/08/04 19:27:47.448|2010/08/04 19:27:47.445 UTC

Event Type: SLC

Message: CSLC042I Process ended when expected - between dMin and dMax. SLC:

bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1.83947193024450767

SLC Instance ID: 83947193024450767

Rule Triggered: Proc duration okay|Proc duration okay - both

Alert: 0|3

Triggering event:

```
<event><alert>|</alert><ruleInstanceId>|</ruleInstanceId><actionId>|</actionId><ruleI
d>|</ruleId><localNode>P</localNode><nodeName>aix534000sp</nodeName><divg>Both</divg><a
ctionsFlag>2010/08/04 19:27:47.465|2010/08/04
19:27:47.432</actionsFlag><dateTime>2010/08/04
19:27:50.000</dateTime><emailFlag>0|0</emailFlag><eventId>83947193024450791</eventId>
```

```
<eventType>4</eventType><fileSize>-1</fileSize><messageId>XSMG252I</messageId><nodeId
>aix534000sp</nodeId><nodeType>1</nodeType><origNode>aix534000sp</origNode><percentCo
mplete>-
```

```
1</percentComplete><processId>28754</processId><processName>sample2</processName><rem
oteNode>hpig4000sp</remoteNode><returnCode>0</returnCode><seqNum>1</seqNum><shortText
>Process completed,
```

```
SNODE:hpig4000sp</shortText><slaFlag>0</slaFlag><submitterId>qatest</submitterId><tra
pFlag>0|0</trapFlag><userOpFlag>0|0</userOpFlag><SUBI>qatest@aix534000sp</SUBI><SBND>
aix534000sp</SBND><STPT>2010/08/04 19:27:50.000</STPT><STRT>2010/08/04
19:27:49.000</STRT><recCat>CAPR</recCat><recordId>PRED</recordId></event>
```

Event XML:

```
<event><ruleInstanceId>|</ruleInstanceId><activityType>P</activityType><sourceEventTi
me>1280950070000</sourceEventTime><groupId>bothStandardSLC1</groupId><divg>Both</divg><
actionsFlag>2010/08/04 19:27:47.448|2010/08/04
19:27:47.445</actionsFlag><actionId>[alert0]| [alert3]</actionId><alert>0|3</alert><da
teTime>2010/08/04
```

```
19:27:47.438</dateTime><emailFlag>0|0</emailFlag><eventId>83947193024450794</eventId>
<eventType>6</eventType><fileSize>-1</fileSize><messageId>CSLC042I</messageId><nodeId
>aix534000sp</nodeId><nodeType>1</nodeType><percentComplete>-
```

```
1</percentComplete><processId>28754</processId><processName>sample2</processName><rem
oteNode>hpig4000sp</remoteNode><returnCode>0</returnCode><ruleId>Proc duration
okay|Proc duration okay -
```

```
both</ruleId><seqNum>83947193024450794</seqNum><shortText>CSLC042I Process ended when
expected - between dMin and dMax. SLC:
```

```
bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1.83947193024450767</shortTex
t><slaFlag>0</slaFlag><slcId>bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.
1.83947193024450767</slcId><slcInstanceId>83947193024450767</slcInstanceId><slcSource
1>bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1</slcSource1><slcSource2>a
ix534000sp.sample2.28754.aix534000sp</slcSource2><slcSourceEventId>83947193024450791<
/slcSourceEventId><trapFlag>0|0</trapFlag><userOpFlag>0|0</userOpFlag></event>
```

```

SLC: bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1.83947193024450767
SLC Instance ID: 83947193024450767
Rule Triggered: Proc duration okay|Proc duration okay - both
Alert: 0|3
Triggering event:
<event><alert>|</alert><ruleInstanceId>|</ruleInstanceId><actionId>|</actionId><ruleId>|</ruleId><localNode>P</localNode><nodeName>aix534000sp</nodeName><dvgtBoth</dvgt><actionsFlag>2010/08/04 19:27:47.465|2010/08/04 19:27:47.432</actionsFlag><dateTime>2010/08/04 19:27:50.000</dateTime><emailFlag>0|0</emailFlag><eventId>83947193024450791</eventId><eventType>4</eventType><fileSize>-1</fileSize><messageId>XSMG252I</messageId><nodeId>aix534000sp</nodeId><nodeType>1</nodeType><origNode>aix534000sp</origNode><percentComplete>-1</percentComplete><processId>28754</processId><processName>sample2</processName><remoteNode>hpig4000sp</remoteNode><returnCode>0</returnCode><seqNum>1</seqNum><shortText>Process completed,
SNOde:hpig4000sp</shortText><slaFlag>0</slaFlag><submitterId>qatest</submitterId><trapFlag>0|0</trapFlag><userOpFlag>0|0</userOpFlag><SUBI>qatest@aix534000sp</SUBI><SBND>aix534000sp</SBND><STPT>2010/08/04 19:27:50.000</STPT><STRT>2010/08/04 19:27:49.000</STRT><recCat>CAPR</recCat><recordId>PRED</recordId></event>
Event XML:
<event><ruleInstanceId>|</ruleInstanceId><activityType>P</activityType><sourceEventTime>1280950070000</sourceEventTime><groupId>bothStandardSLC1</groupId><dvgtBoth</dvgt><actionsFlag>2010/08/04 19:27:47.448|2010/08/04 19:27:47.445</actionsFlag><actionId>[alert0]| [alert3]</actionId><alert>0|3</alert><dateTime>2010/08/04 19:27:47.438</dateTime><emailFlag>0|0</emailFlag><eventId>83947193024450794</eventId><eventType>6</eventType><fileSize>-1</fileSize><messageId>CSLC042I</messageId><nodeId>aix534000sp</nodeId><nodeType>1</nodeType><percentComplete>-1</percentComplete><processId>28754</processId><processName>sample2</processName><remoteNode>hpig4000sp</remoteNode><returnCode>0</returnCode><ruleId>Proc duration okay|Proc duration okay - both</ruleId><seqNum>83947193024450794</seqNum><shortText>CSLC042I Process ended when expected - between dMin and dMax. SLC:
bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1.83947193024450767</shortText><slaFlag>0</slaFlag><slcId>bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1.83947193024450767</slcId><slcInstanceId>83947193024450767</slcInstanceId><slcSource1>bothStandardSLC1.bothDuration1.aix534000sp.sample2.true.1</slcSource1><slcSource2>aix534000sp.sample2.28754.aix534000sp</slcSource2><slcSourceEventId>83947193024450791</slcSourceEventId><trapFlag>0|0</trapFlag><userOpFlag>0|0</userOpFlag></event>

```

The following event elements may be seen in events included in an SLC Debug Report:

Event Element	Description
actionId	Action performed when the event was processed
actionsFlag	Tells when the event was processed by the Rule service
activityType	The type of SLC component that generated the event: <ul style="list-style-type: none"> WF—Workflow M—Milestone S—Process Step P—Process
dateTime	Time the event was generated (values are in UTC)

Event Element	Description
destFile	Destination file name watched for by this SLC instance
groupId	SLC name
nodeId	Server name watched for by this SKC instance
ruleId	Rule triggered by the event
slcId	For Wildcard SLCs, formed by concatenating the SLC Name, the SLC schedule name, the server name specified, the destination file name specified, and other values
slcSourceEventId	Event ID of the event that caused this SLC event to be generated (not present in the sample debug report provided because it was the lack of process events that caused these SLC events to be generated)

Update Information for Multiple Servers

If the same user ID and password are used to attach multiple servers to Control Center, you may want to use the technique described in this chapter to perform a mass change of all passwords to avoid having to individually update each server. This chapter describes how to create a template to use with a script and then running RUNBATCH to do the password change en masse. For a detailed explanation of the sample script provided in Control Center (*script.txt*) and templates, see the *Creating Multiple Objects* appendix in the *Sterling Control Center System Administration Guide*. For another example using RUNBATCH, see *How Can I Use RUNBATCH To Create Schedules for an Hourly Process?* on page 138.

How Can I Do a Bulk Update of the Passwords Used by Control Center for Monitored Servers?

When you add a server to monitor, you specify a User ID and Password for Control Center to use to log onto the server. Instead of individually changing the password property for each server, you can follow this basic procedure:

1. Copy the xml definitions of all servers whose passwords you want to update from the *installation directory/conf/services/nodes* folder to the template folder, and change all *xml* extensions to *tmp*.
2. Edit these new template files and insert a password variable by replacing the text between `<password>` and `</password>` with `&pw;`.
3. Create a script that includes the following three lines for each template supplying your data in the italicized items:

```
copy <template filename (without the .tmp extension)>  
&pw;=<new password value>  
SERVERUPDATE
```

4. Type **runBatch** *hostname port userid password scriptname* (Windows) or **runBatch.sh** *hostname port userid password scriptname* (UNIX).

Here is a sample template created from the xml definition for a server named *SFG75039* (the server ID is italicized below). The sample template is named *SFG75039.tmp*. The encoded version of the encrypted password was replaced with *&pw*.

```
<service>
  <apiPort>1463</apiPort>
  <applyMetadataRules>>true</applyMetadataRules>
  <connectionTimeout>30</connectionTimeout>
  <contact/>
  <contactEMail/>
  <contactName/>
  <contactPhone/>
  <desc/>
  <hostName>SFG</hostName>
  <id>SFG75039</id>
  <ignoreServerProvidedTZ>>false</ignoreServerProvidedTZ>
  <licenseExpiryWarning>30</licenseExpiryWarning>
  <maxCompletedProcs>100</maxCompletedProcs>
  <notManaged>>false</notManaged>
  <notMonitored>>false</notMonitored>
  <password>&pw;</password>
  <pollingRate>5</pollingRate>
  <protocol>TCPIP</protocol>
  <serverMetaData1/>
  <serverMetaData10/>
  <serverMetaData2/>
  <serverMetaData3/>
  <serverMetaData4/>
  <serverMetaData5/>
  <serverMetaData6/>
  <serverMetaData7/>
  <serverMetaData8/>
  <serverMetaData9/>
  <serverTimeZoneId>America/Chicago</serverTimeZoneId>

  <serviceClass>com.sterlingcommerce.scc.agent.services.node.cd.CDNodeService</
serviceClass>
  <serviceId>SFG75039</serviceId>
  <serviceType>1</serviceType>
  <skipProcessStepInfo>>false</skipProcessStepInfo>
  <sourcePort/>
  <synchObjectTime>SYSTEM</synchObjectTime>
  <tracingEnabled>>false</tracingEnabled>
  <userId>cduser</userId>

</service>
```

Here is a sample script which tells RUNBATCH to copy the *SFG75039* template, substitute the new value (*newpassword*) for the specified password variable, and update the xml definition for the *SFG75039* server. Note that the SERVERUPDATE command tells RUNBATCH to only update the information for the server versus create the xml definition from scratch.

```
copy SFG75039
&pw;          = newpassword
SERVERUPDATE
```

The last step is to run *runBatch.bat* file from the *installation directory/bin* folder to run the sample script.

A

- actions
 - creating
 - generate an alert 33
 - generate an SNMP trap 33
 - to release a
 - Process 8, 16, 17, 18, 19, 22, 23, 24, 25, 28
 - to send an e-mail 31
 - setting up 7, 21, 27
- alert
 - how to generate 33
- automated reports 91

C

- command
 - operating system 35
- correlating milestones 134

E

- e-mail
 - creating an action to send 31
- examples
 - reports 91

M

- monitor
 - out-of-compliance 77
 - rest time 77
 - server on regular basis (messages) 77
- multiple servers
 - updating information for 169

O

- operating system command 35
- out-of-compliance events
 - monitor 77

- troubleshooting 153

P

- passwords
 - doing a bulk update 169
- Process
 - creating an action to
 - release 8, 16, 17, 18, 19, 22, 23, 24, 25, 28

R

- release
 - Process
 - creating an action
 - to 8, 16, 17, 18, 19, 22, 23, 24, 25, 28
- reports
 - automated 91
- rules
 - CE
 - remote connect delete 63
 - remote connect directory list 64
 - remote connect disconnection 65
 - remote connection add 58
 - remote connection request 57
 - CE offline add 59
 - CE UNIX
 - autoconnect 60, 71, 72
 - daemon goes down 53
 - daemon starts 52
 - daemon stops 51
 - offline extraction 66
 - CE z/OS
 - autoconnect 67, 68, 69, 70
 - copy completes 41
 - copy completes unsuccessfully 42
 - file was delivered to a consumer's mailbox 80
 - license expiring 73
 - notify Producer if file does not arrive 81
 - Process completes 39
 - Process goes to Held In Error 50
 - Process has problem 40

Index

- send an e-mail if database problem 45
- server down alert 46, 48
- server goes down 43
- server starts 44
- setting up 37
- specifying more than one criterion 75
- Sterling File Gateway route fails 78
- Sterling Integrator
 - notify delivery to mailbox 79
- Sterling Integrator adapter
 - goes down 56
 - is enabled 55
 - stops 54
- troubleshooting 83

S

schedules

- prevent rules from triggering 87
- setting up 87

setting up

- actions 7, 21, 27
- rules 37
- schedules 87
- SLCs 93

SLCs

- alert when related CE batches arrive 106
- create schedules 138
- file transfer 126
- hourly Process 127
- Process
 - hourly 127
 - start on time 119
- Process fails to complete 81
- Process runs too long 102, 105
- reduce number of events 124
- related Processes
 - fail to complete 118
 - fail to start or complete on time 120
 - may not complete on time 121
 - run too long 112
- RUNBATCH 138
- setting up 93
- Sterling Integrator
 - know when a file was not delivered on time 132
 - notify Producer if Consumer does not pick up 134
- tips 144
- troubleshooting 143
- what events generated 80, 97

- which type to use 101, 134
- why need 96, 132

- SLCs and rules 78, 94, 169
 - when use 79, 95

- SNMP trap
 - how to generate 33

T

time zones

- troubleshooting SLCs 152

troubleshooting

- out-of-compliance 153
- SLCs 143

V

- variables 35