SNMP ColdFusion Event Gateway Documentation

This document provides instructions on the configuration and use of the SNMP Event Gateway.

Configuring the SNMP Event Gateway

Before you can use the SNMP Event Gateway you must first configure it in your ColdFusion server.

This is a fairly simple process. You will need the following things:

- The SNMPGateway.jar file
- The SNMP4J.jar file

Additionally, you will create a configuration file.

First, copy the SNMP4J.jar under your ColdFusion server's lib directory. If you're running on JRun this will be under the JRun4 folder. On the standalone version this directory is under your ColdFusion root directory.

Secondly, copy the SNMPGateway.jar file into your ColdFusion server's ***gateway*** lib directory. If you're running on JRun this will be under your server instance at /cfusion-ear/cfusion-war/WEB-INF/cfusion/gateway/lib/. For example, /JRun4/servers/cfusion/cfusion-ear/cfusion-war/WEB-INF/cfusion/gateway/lib/. On the standalone install this will be under /gateway/lib under your ColdFusion root directory.

Next, you will need to create a configuration file. You should have a sample file called SnmpGateway.cfg. You can place this file wherever you want. You'll need a new version of this file for each SNMP gateway instance you create.

In the configuration file you will be able to set the following options:

listener_address

The listener_address configuration value is used to specify a particular IP address that the SNMP Event Gateway listens on. The default value of 0.0.0.0 indicates that all IPs will be used. You may want to change this if you create more than one instance so that the instances do not conflict by trying to listen on the same IP address and port.

listener_port

The listener_port configuration value is used to specify a particular port that the SNMP Event Gateway listens on. The default value is 1162. You may want to change this if you create more than one instance so that the instances do not conflict by trying to listen on the same IP address and port.

changeFunction

The changeFunction configuration value specifies the name of a function that the event gateway will call a CFC you will create. (More on the CFC later in the documentation.) The default value is onEvent. It's not recommended to change this.

logging

This indicates if additional information is to be logged into the ColdFusion instance's out log. This is not implemented and should be ignored.

Configure ColdFusion

To use the Event Gatway you'll need to create both a Gateway Type and a Gateway instance in your ColdFusion Administration interface. To do this, log in to your ColdFusion Administrator and click Event Gateways > Gateway Types. This will show the Add / Edit ColdFusion Event Gateway Types form. Provide the values specified below:

Type Name and Description: We suggest you provide the text "SNMP Event Gateway". However you can set this to whatever you like.

Java Class: This must be "com.esc.msu.SnmpGateway"

Startup Timeout: It is recommended that you leave this at the default of 30 seconds.

Click to add the type and confirm that it's added to your list of configured gateway types.

Now create a new Gateway Instance by clicking "Event Gateways > Gateway Instances". In the resulting form create a new instance by providing the values specified below:

Gateway ID: This can be anything you want but must be unique. We suggest simply providing an id of "snmp". All of the examples in this documentation will assume that's the ID you're using.

Gateway Type: Select the instance type you created in the previous step. We recommended you call this SNMP Event Gateway.

CFC Path: This is the fully qualified path to the CFC that will handle events generated by this Gateway. We'll discuss the details of this CFC later in this document. For now, create an empty CFC somewhere and provide the path to tha CFC.

Configuration File: Provide the fully qualified path to the configuration file you created.

Startup Mode: For production deployments we recommend you set this to always. Otherwise, choose whichever is most appropriate.

Click Add Gateway Instance. This will create the instance but it ***will not start it for you automatically***. If you selected automatic startup it *will* restart when you retart ColdFusion. You should see the instance appear in the list of Gateway instances. Click the green start button to start the Gateway instance. The page should refresh and you should see a status of "running" for your new SNMP Gateway instance.

Congratulations, you've configured the gateway!

Creating your Gateway CFC

To use the SNMP Event Gateway you'll need to create a new CFC with a function that can receive events from the Event Gateway. This CFC must have a function named onEvent (or whatever you named this method in your configuration file). This onEvent function must accept one argument named CFEvent of type struct. Here's some dummy code to get our started:

```
<cfcomponent hint="I this is a sample CFC that handles SNMP Events">

      <cffunction name="onEvent" access="public" hint="I handle
      incomming events from an event gateway." output="false"
      returntype="void">
            <cfargument name="CFEvent" hint="I am the cfevent
            structure" required="true" type="struct" />
            <cfset var dump = 0 />

      </cffunction>

</cfcomponent>
```

This CFC will be notified by the gateway when an SNMP event occurs. The Gateway will call the onEvent function and pass in a structure of data. The structure will have the following values:

cfcmethod: This is the name of the method being called. This should be onEvent by default.

cfcPath: This is the path to the CFC being called. You set this in the Gateway Instance configuration.

gatewayId: This is the ID of the gateway instance as configured in the ColdFusion administrator.

gatewayType: This is the gateway type as you specified when creating the type.

originatorId: This is a standard field provided by ColdFusion and should be disregarded.

data: This is another struct which contains the relevant data about the SNMP event. The elements in this structure are as defined below:

> **CommunityString:** This is the SNMP CommunityString
>
> **EventTimeStamp:** This is the event's time stamp
>
> **EventType:** This is the SNMP event type.
>
> **EventVarbinds:** This is a structure of varbinds in the SNMP Event Gateway. There will be one element per varbind in the structure.
>
> **EventVersion:** This is the type of event.
>
> **RequestId:** The SNMP request ID.
>
> **Sender:** This is the IP address and port of the sender of the SNMP event.
>
> **TimeReceived:** This is the data and time the event was received.

Using the information provided you can take any action you wish within your CFC.

Announcing SNMP Events

Event Gateways in ColdFusion can make use of what are known as helpers. To send SNMP events you will use the SNMP Event Gateway "helper" component.

To get the helper you will need to use the ColdFusion function GetGatewayHelper(). Pass this function the name of the gateway instance you're getting the helper for. For example:

```
<!---
      Get a handle on the snmp event gateway helper.
      Note that the name of the gateway configured is expected to be
      "snmp"
--->
<cfset snmpHelper = GetGatewayHelper("snmp") />
```

Once you have the SNMP Helper you can call functions on this object. (Note: this is a Java object and not a ColdFusion component.) The functions available on this object are as specified below:

SnmpGatewayHelper Methods

SnmpGatewayResponse = createCredentials(string target, string community)

This function creates a credentials object. As with the helper, this is a Java object. The credentials object is passed to the get function on the helper object. This function accepts the following arguments:

Target: This is a string value indicating the IP address of the target SNMP agent.

Community: This is the community string to use in requests made on the target SNMP Agent .

This function returns an SnmpGatewayResponse object.

Example Code:

```
<cfset snmpHelper = GetGatewayHelper("snmp") />

<!--- create the credentials --->
<cfset credentials = snmpHelper.createCredentials("192.168.1.1",
"public") />
```

SnmpGatewayResponse = get(SnmpGatewayCredentials cred, Array vbs)

The get function invokes an SNMP get request.

Cred: This is the SnmpGatewayResponse object created by calling the createCredentials() function.

Vbs: This is a ColdFusion array of varbinds to include in the get request.

Example Code:

```
<cfset snmpHelper = GetGatewayHelper("snmp") />

<!--- create the credentials --->
<cfset credentials = snmpHelper.createCredentials("192.168.1.1",
"public") />

<!--- create the varbinds --->
<cfset varbinds = ArrayNew(1) />

<cfset ArrayAppend(varbinds, "1.3.6.1.2.1.1.3.0") />
<cfset ArrayAppend(varbinds, "1.3.6.1.2.1.1") />
<cfset ArrayAppend(varbinds, "1.3.6.1.2.1.1.5.0") />

<!--- do a get request --->
<cfset response = snmpHelper.get(credentials, varbinds) />
```

SnmpGatewayResponse = getNext(SnmpGatewayCredentials cred, Array vbs)

The getNext function invokes an SNMP GetNext request.

Cred: This is the SnmpGatewayResponse object created by calling the createCredentials() function.

Vbs: This is a ColdFusion array of varbinds to include in the GetNext request.

Example Code:

```
<cfset snmpHelper = GetGatewayHelper("snmp") />

<!--- create the credentials --->
<cfset credentials = snmpHelper.createCredentials("192.168.1.1",
"public") />

<!--- create the varbinds --->
<cfset varbinds = ArrayNew(1) />

<cfset ArrayAppend(varbinds, "1.3.6.1.2.1.1.3.0") />
<cfset ArrayAppend(varbinds, "1.3.6.1.2.1.1") />
<cfset ArrayAppend(varbinds, "1.3.6.1.2.1.1.5.0") />

<!--- do a get request --->
<cfset response = snmpHelper.getNext(credentials, varbinds) />
```

SnmpGatewayResponse Methods

String = getDuration()

The getDurration() function returns the duration between the time the request was sent and the response received in milleseconds. The word “milleseconds” is included. To remove this simply use the ColdFusion val() function.

Int = getErrorIndex()

The getErrorIndex() function indicates which of the requested varbinds is associated with the error status. The first varbind is 1.

Int = getErrorStatus()

The getErrorStatus() function returns the error status number.

String = getErrorStatusText()

The getErrorStatusText() function returns a text associated with the error status.

String = getRequestType()

Struct = getRequestType()

The getRequestType() function gets the request type of the original request sent for this response.

String = getStart()

The getStart() function returns the date and time the request was sent.

String = getSynopsis()

The getSynopsis() function returns a description of the response in text using the other functions in this object.

String = getTarget()

The getTarget() function returns the target of the SNMP request and response.