Automation of two HP SIM 5.1 or greater servers to provide Failover Event Handling

In very large environments, one HP Systems Insight Manager (HP SIM) may not provide all the coverage needed for all the devices or for disaster recovery (DR). It might be desirable to have two HP SIM servers covering for each other. In this scenario, we will create a set of Event Handlers which allow the two HP SIM (Central Management Server or CMS) servers to specifically watch each other and take over coverage for the other's Event Notification in the event of failure of one CMS.

The document describes how to create an Event Handler using the CMS Command Line tool mxtask, one of many CLI tools that come with HP Systems Insight Manager and are documented in the Information Library at:

http://h18013.www1.hp.com/products/servers/management/hpsim/infolibrary.html.

The goal is to have two HP SIM servers that manage Event Notification for local devices, say, City A and City B. In normal operation, City manages only City A servers and vice versa for City B. In the situation that the HP SIM server in City A fails, it is desirable to have City B automatically detect this and run a task the would enable it to handle alerts on City A server in addition to its own population. The basic configuration is that each CMS has a full population of all devices being monitored, both from City A and B. This document describes a method to achieve that failover.

The steps are:

1. Create two Event Collections using the customize lists option, one for City A events, the other for City B events.

2. Create an Event Handlers on Server A and B that watch each other and run a custom task that enables a new Event Handler for the other cities devices when the CMS in question dies.

3. Create an Event Handlers on Server A and B that watch each other and run a custom task that removes the event handler for the other cities devices when the CMS comes back online.

Figure 1: Normal Operation

CMS A processes events on servers in A's territory, CMS B processes events from servers in B's territory. A Watch Dog Event Handlers at each CMS watches for the other CMS to become reachable or unreachable and creates or deletes event handler tasks.
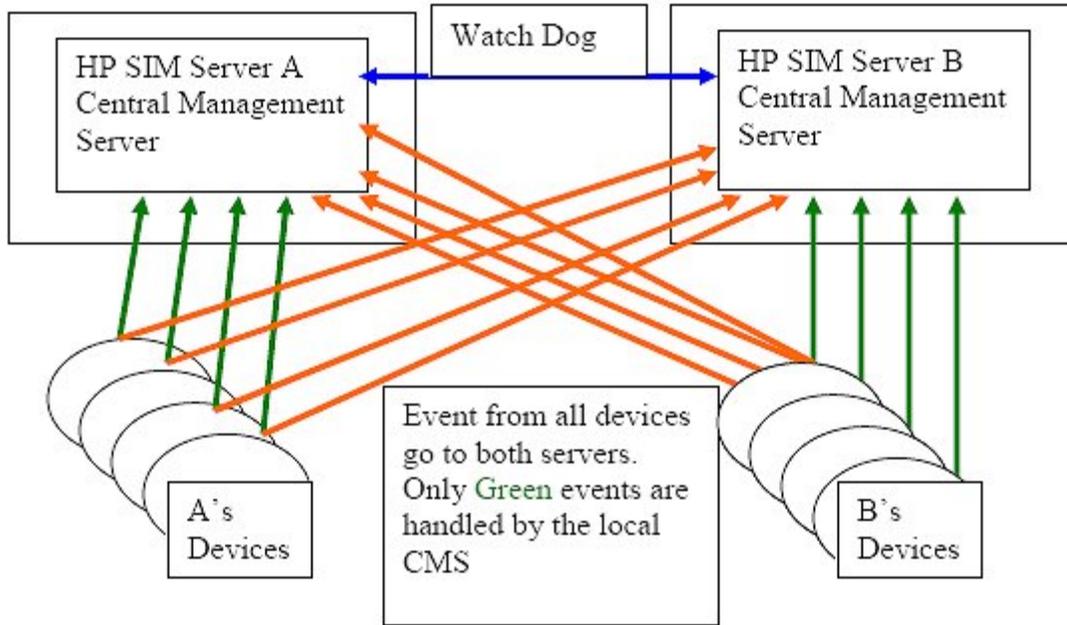
Figure 2: Server B is Unavailable
When Server B dies, then Server A processes events for Servers in B's territory until Server B comes back, because all agents are configure to send events to both server, not changes are needed to agents for achieve failover.
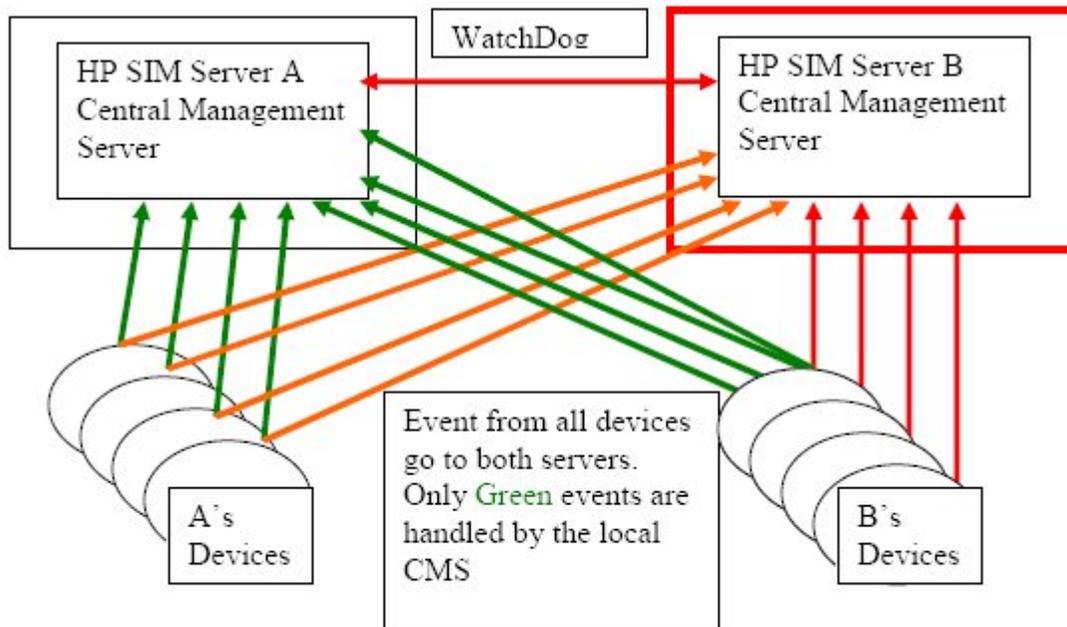

Both CMSes have a full population of all devices in both territories, events from CMS A are active only for A's devices; likewise for B's. All agents send events to both CMSes. Events received by a CMS for devices in the others territory are ignored and/or purged until partner CMS is determined to be unavailable. See Appendix B for instruction on purging events.

Creating a Custom Events List

You must create a Custom Event List for the events that you are interesting in handling. In this example, custom list called "MRO Events" which consisted of specific hardware related events on specific servers.

To create a custom collection or list:

1. Click on Customize (see the red oval), the Customize Collections screen appears. In the drop down list, select Events (see green oval shown) then click **New**.
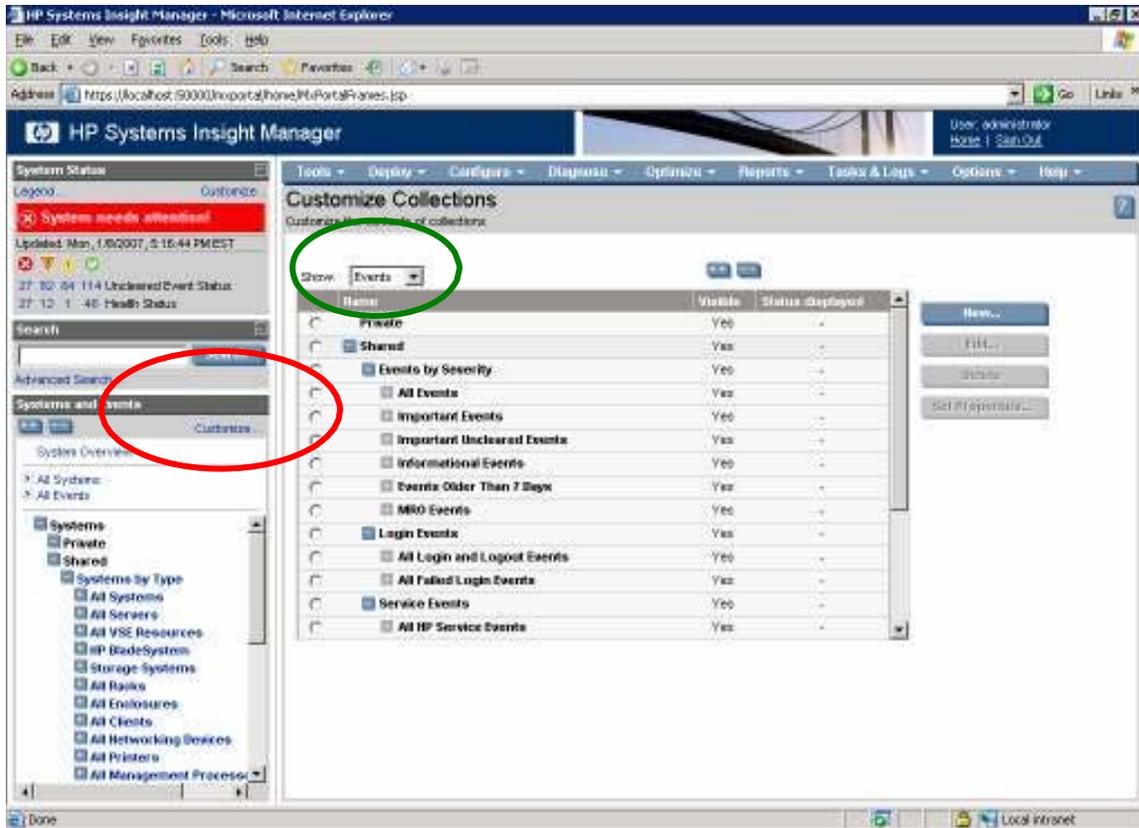
2. Wait for a minute after clicking **New** and notice the New Collection at the bottom of the screen. Select **Choose members by Attribute**.

3. Select **Event Type is** and select the set of events you are interested in as shown below. Click **Add** to add other event criteria.
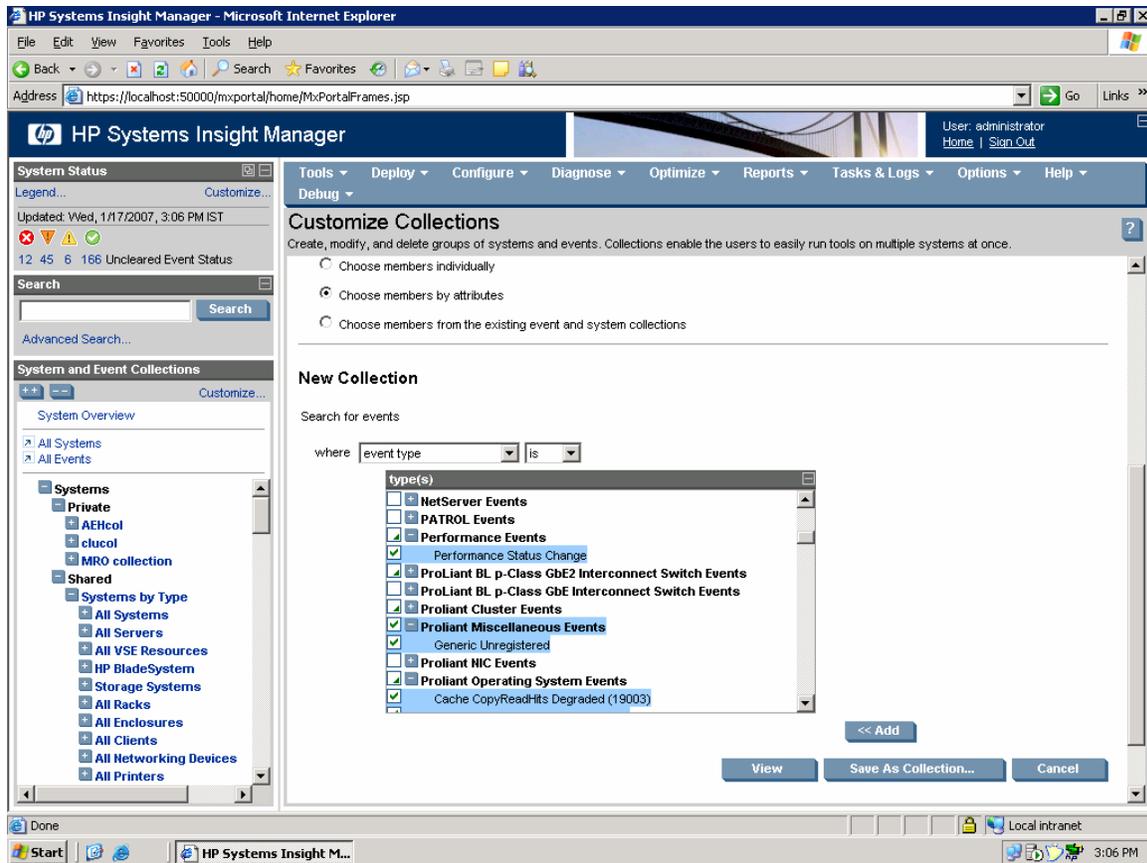
4. Click **Save As** to save the Collection and give it a name. For example, you could enter the name **MRO Events**, select **Events By Severity** from the **Existing Collection** dropdown list, and click **OK** to save.
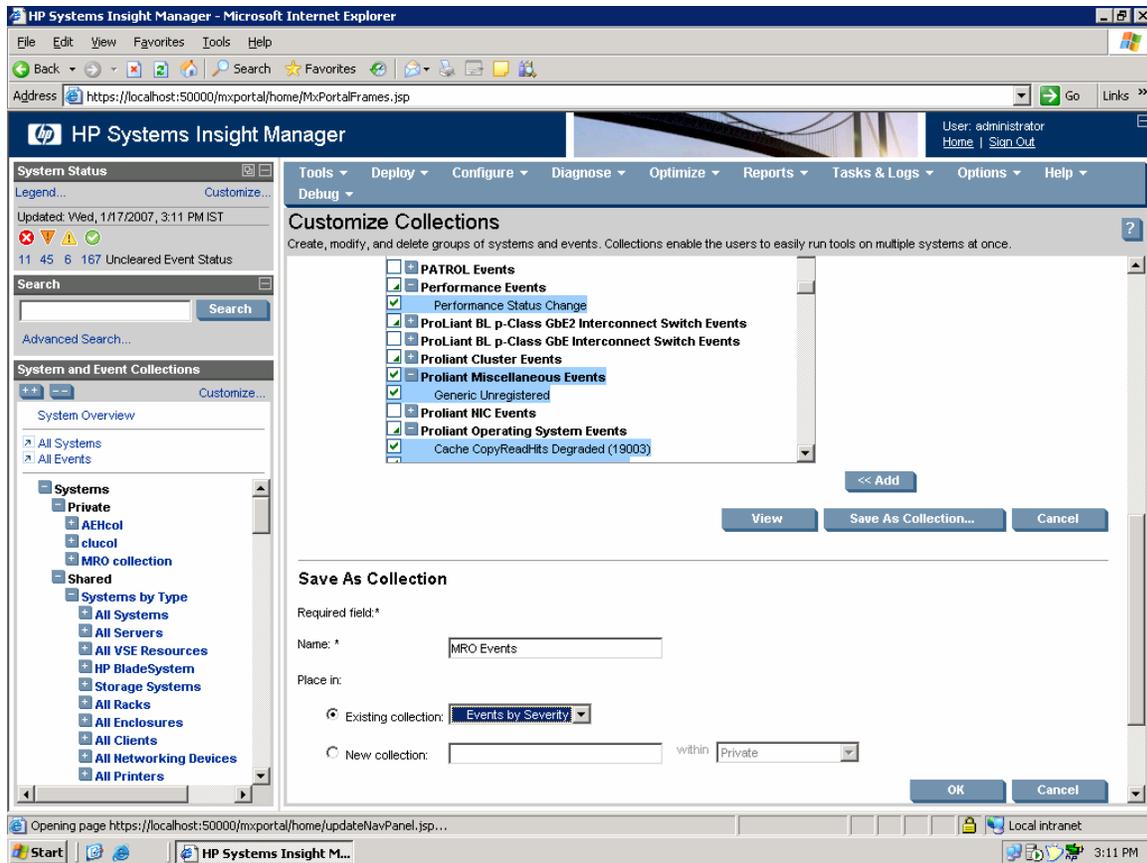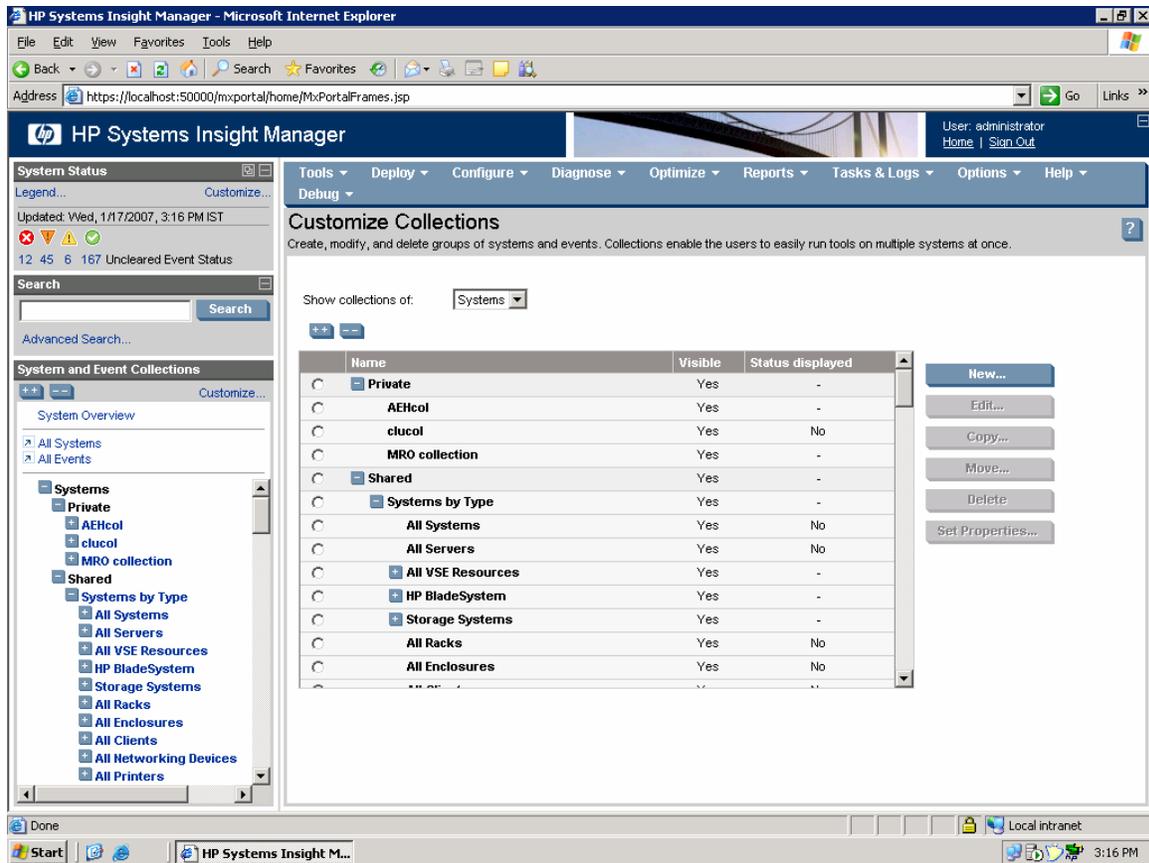
Creating a Custom Systems List

Use Customize link to create system collection.

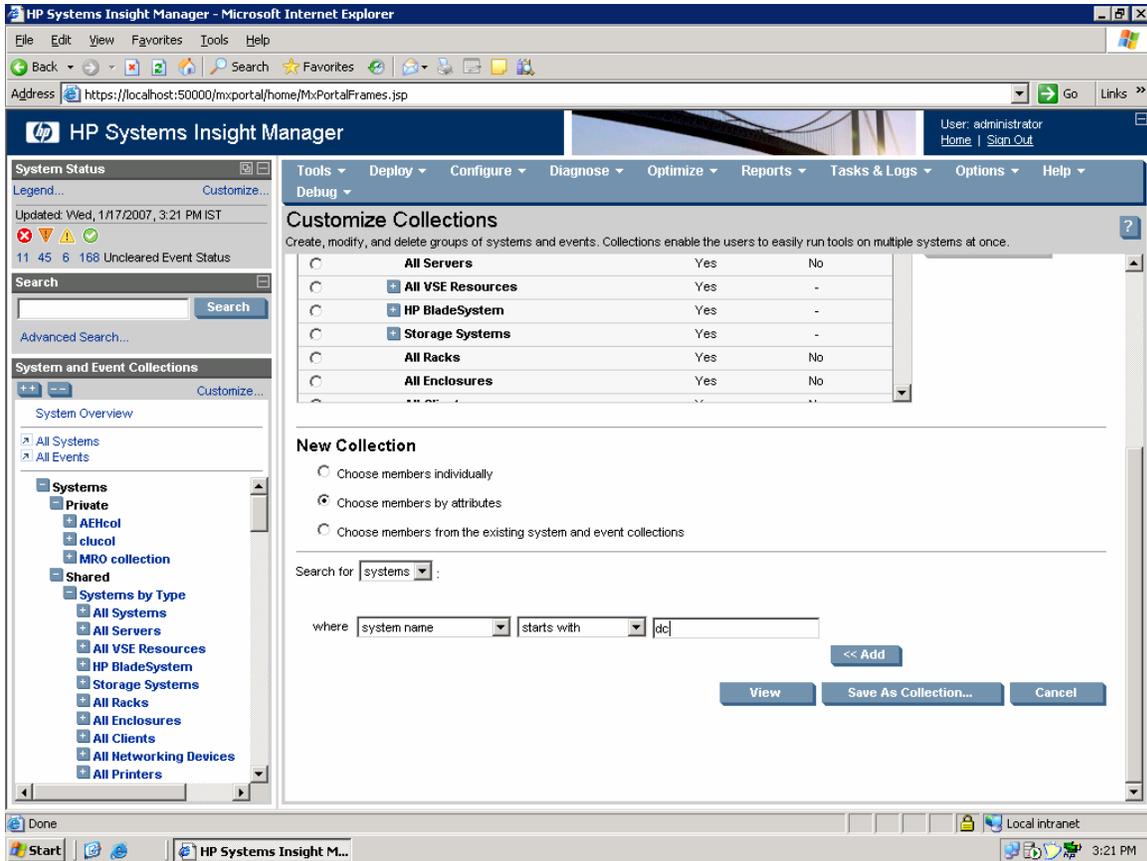To create a custom system collection or list:

1. Click on **Customize** (see the red oval). The **Customize Collections** screen appears. Select **Systems** (see green oval) from the dropdown list and click **New**.

2. After waiting a minute after clicking **New**, the **New Collection** appears at the bottom of the screen. Select **Choose members by Attribute**.
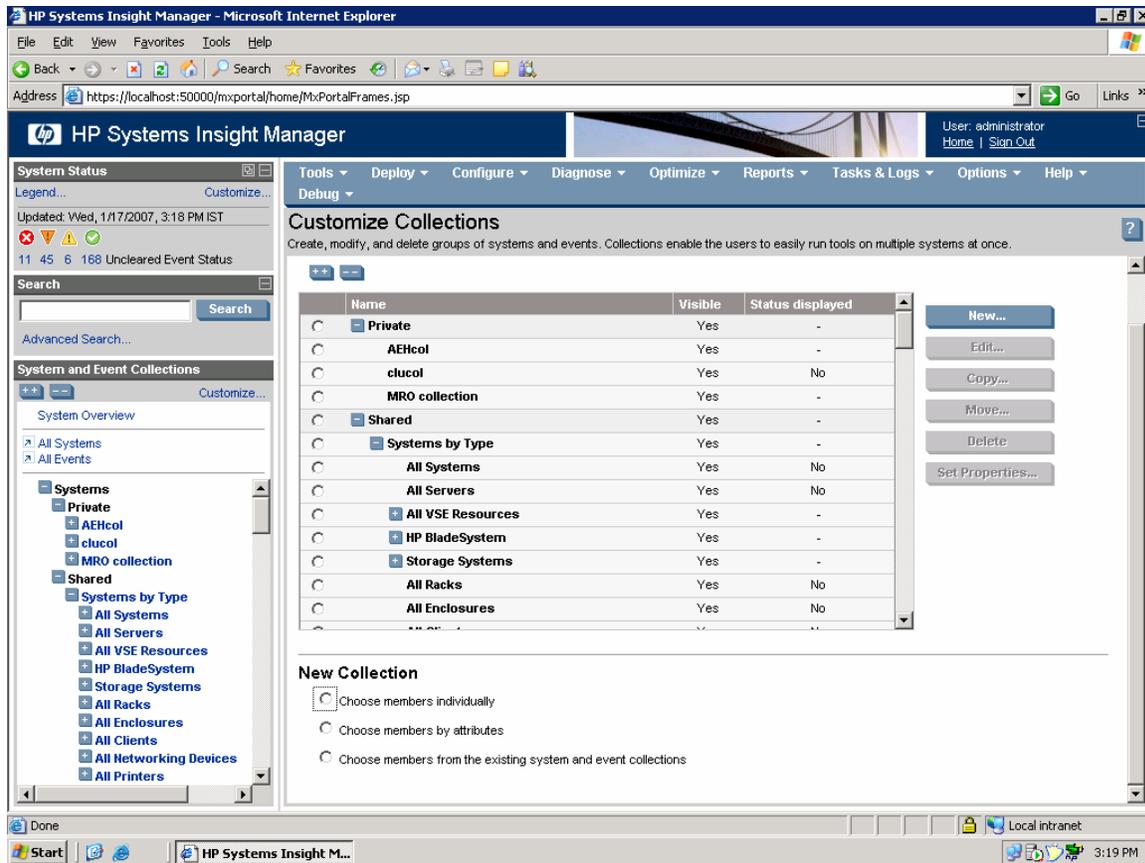
3. From the dropdown list under **New Collection**, select the set of devices to create a system collection where it can be **Server Name Starts With** (see red oval below). In this case the letters **dc** – so we are looking for all servers whose names start with the letter **dc**. Fit the naming scheme to your environment or with any other system criteria.

4. Create two of these collections, in this case one for City A and one for City B.

5. Click **Save As** to save the collection and give it a name. In the example, the name **CITYB Systems** was used. Click **OK** to save the settings. This collection can be used to create a handler.
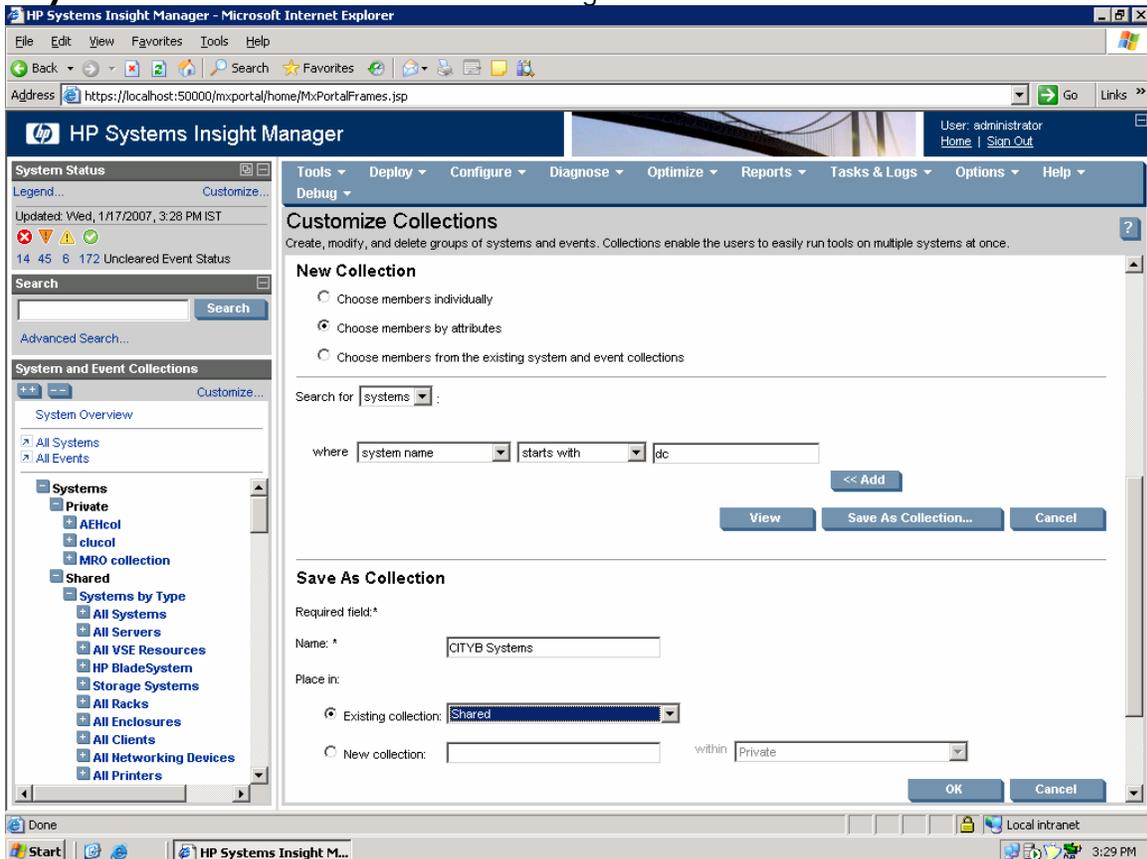
Creating a Custom Event-System Combo collection

You must create a Custom Event -System combo collection for the events that you want to include. Use the two collections shown above, Events ("MRO Events") and Systems ("CITYB Systems") collection to create a new Event- System combo collection.
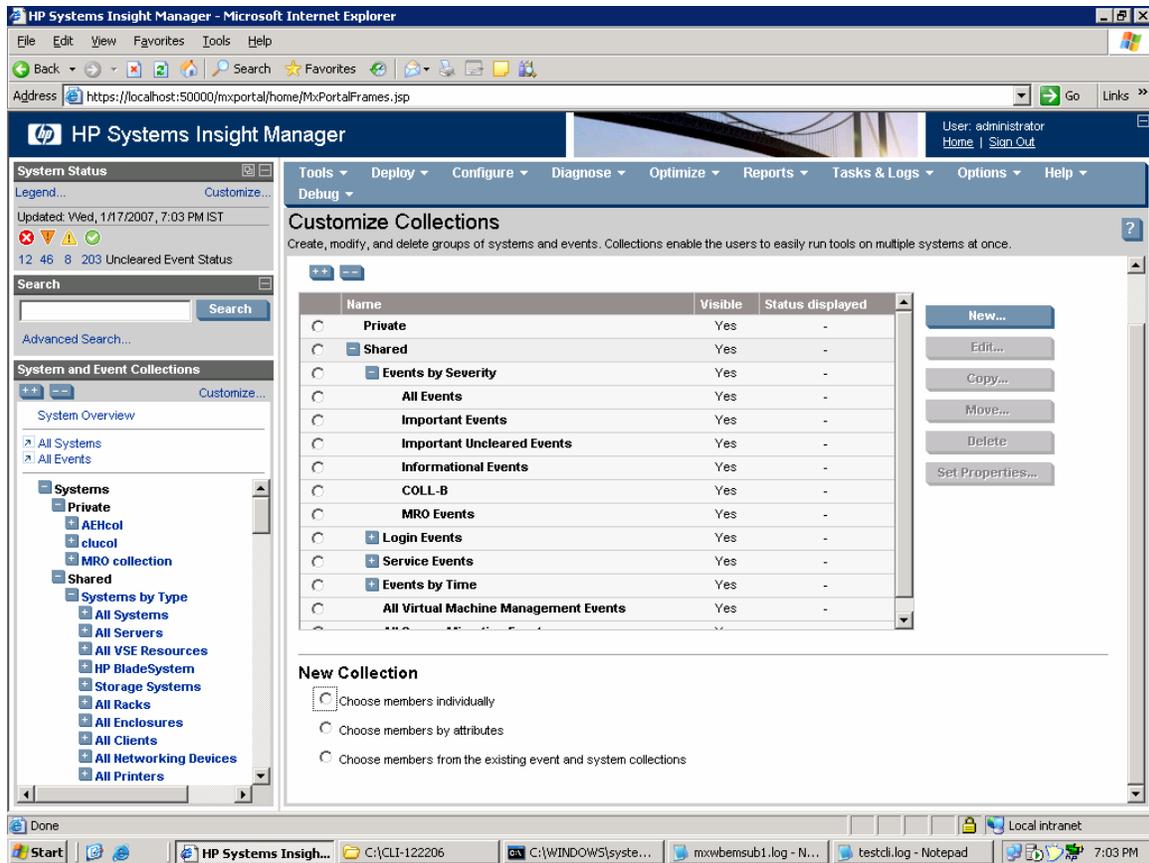
To create a custom combo-collection:

1. Click **Customize** (see the red oval below). The **Customize Collections** screen appears. From the dropdown list, select **System** (see green oval) and click **New**.
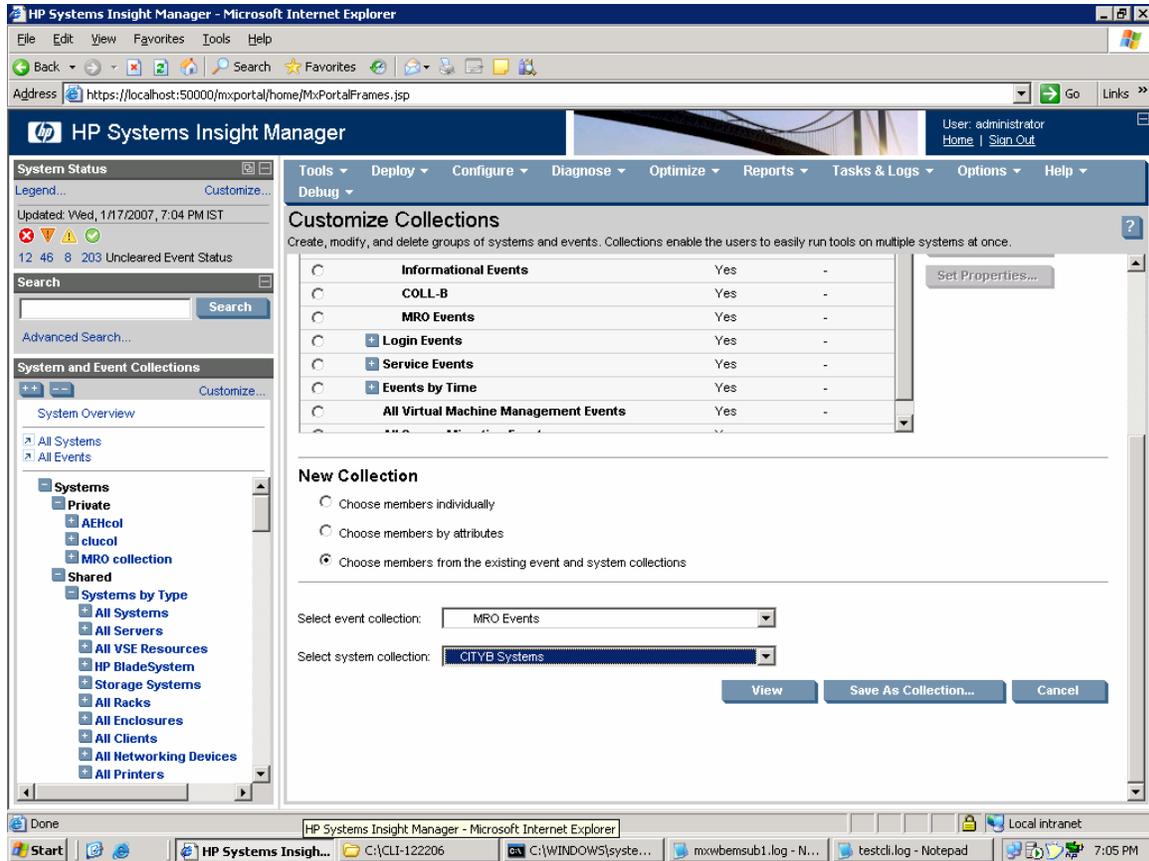

2. After waiting a minute after clicking **New**, select **Choose members from the existing system and event collections** under the **New Collection** section.

3. Under **New Collection**, select newly created **MRO Events** and **CITYB Systems** from the events and systems collection list.

4. Click **Save As** to save the collection and give it a name. In this example, the name **CITYB Event Combo** is used. Click **OK** to save the settings.
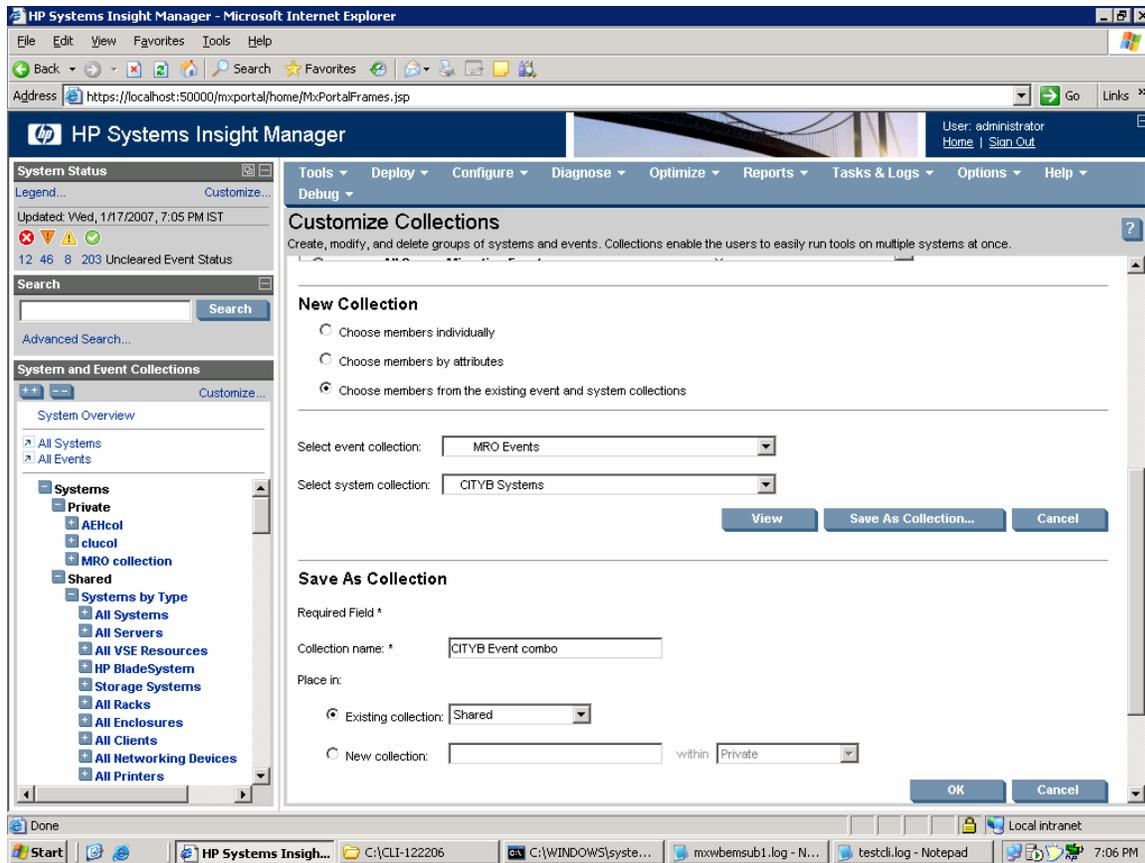
You must have at least two of these collections, one for City A's systems and one for City B's systems. For devices in City A, you must create a normal event handler for those devices in City A using the above Event System combo collection and it is enabled all the time. Similarly, for City B, you use the **Options -> Automatic Event Handing -> New** menu options to set up this standard notification on City B CMS

To handle the transient nature for turning the event handler on and off for the other CMS's devices, we use a Custom tool, which creates the handler for other CMS devices only when that CMS becomes unavailable.

Creating the custom tool

This tool adds an event handler for the devices that no longer have their CMS to perform notification.

1. Select **Tools -> Custom Commands -> New Custom Command** to create a new tool.
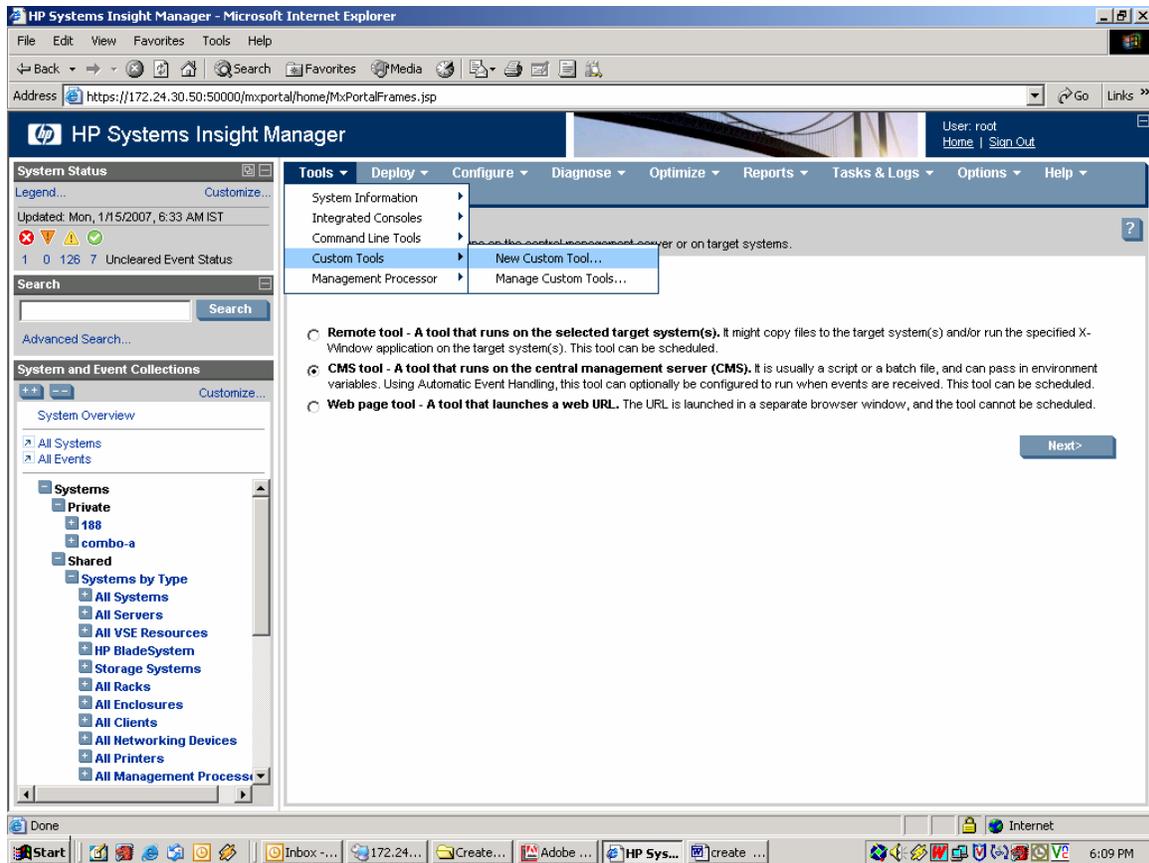
2. From the **New Custom Tool** screen, there are three types of tools. Select **CMS tool** as shown. Click **Next**.

Creating the tool definition

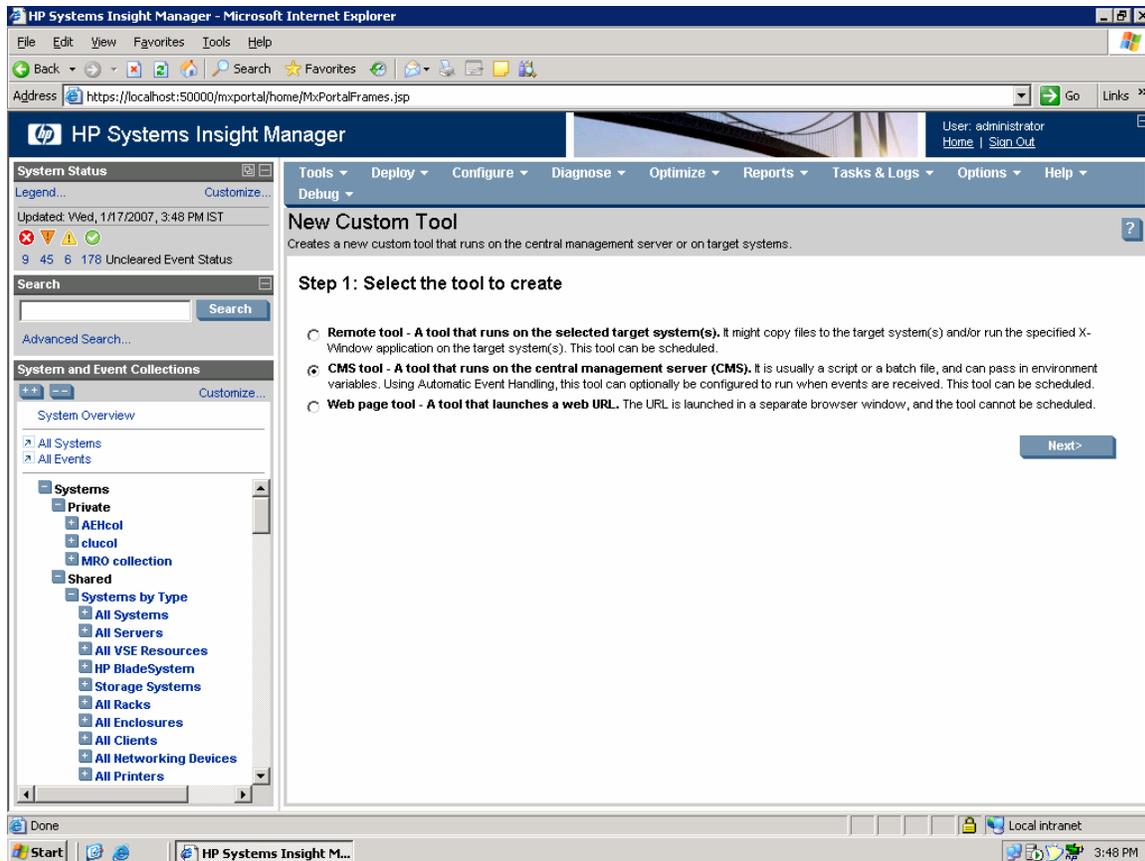Enter the required name (for example, AddHandler), command file, and parameters. The path names must be in DOS 8-dot-3 format. In this example, the command path is the location of the `MXTASK` command. `MXTASK` is used to create a new event handler using an XML file for the rules.
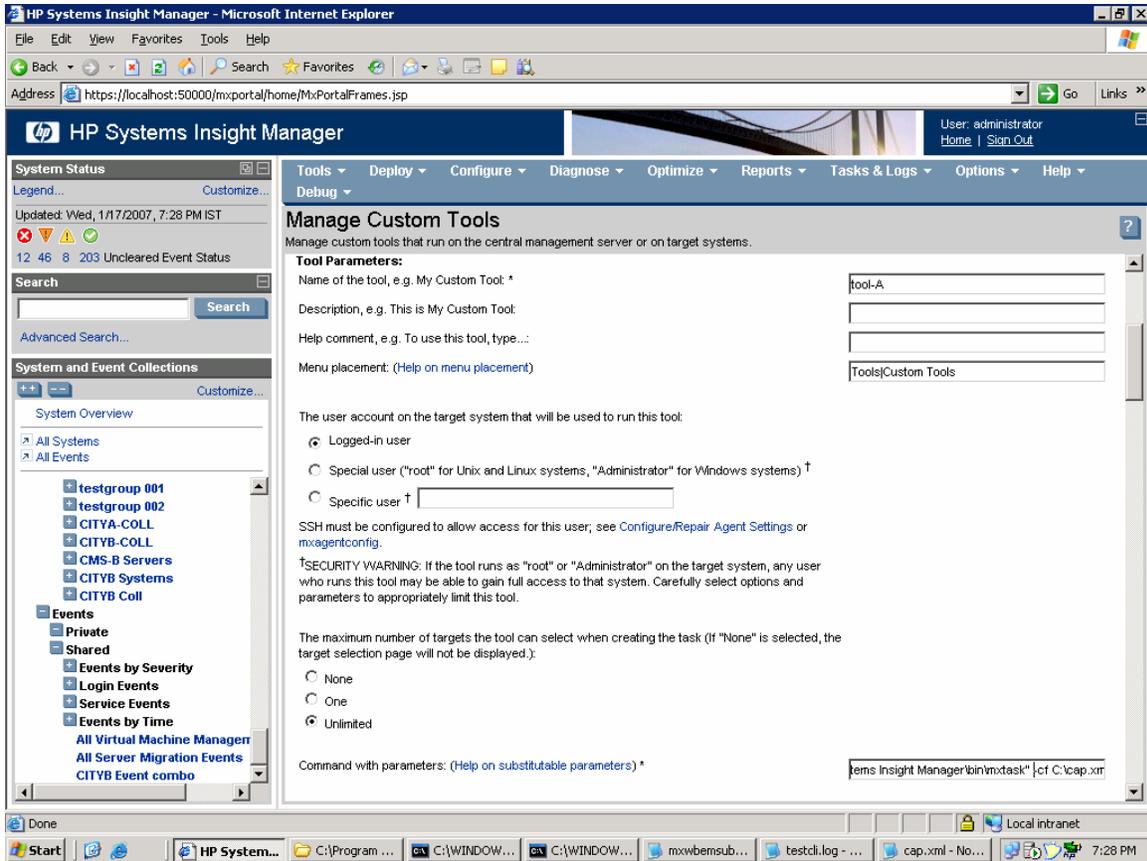
In this example, the command file path is `/opt/mx/bin/mxtask` (Linux)

`C:\Program Files\HP\Systems Insight Manager\bin\mxtask` (Windows).

The parameters are: `-cf  /root/cap.xml (Linux)` , `-cf C:\cap.xml`

The `-cf` command tells mxtask to create a task from a file called *cpa.xml* in XML format. See the appendix for the syntax of the XML file.

Tricks: The simplest way to retrieve the XML file is to use the GUI to create an event handler (Appendix A) that you want and then use the `mxtask -lf myEventHandler >> cap.xml` to capture the content and syntax of the event handler. After you have the syntax correct, delete the task using `mxtask -r MyEventHandler` (for this example, the event handler is named **ClaudiasEvents**). There is no **Modify Task** command so manipulation of the handler if done by creating it and removing it as needed. The path and parameters for this task is shown below.

3. Enter the following command:
   ```
   C:\Program Files\HP\Systems Insight Manager\bin\mxtask" -cf C:\cap.xml
   ```
4. Click **OK** to create the tool.
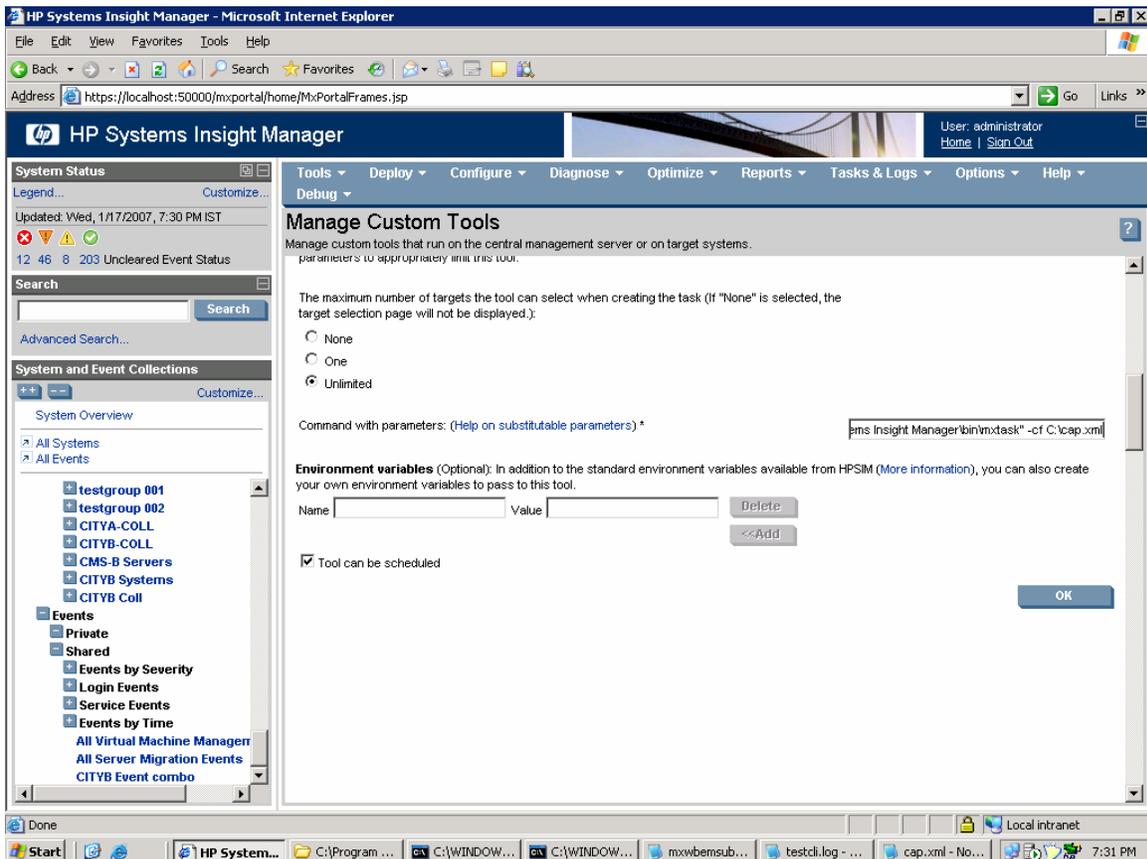
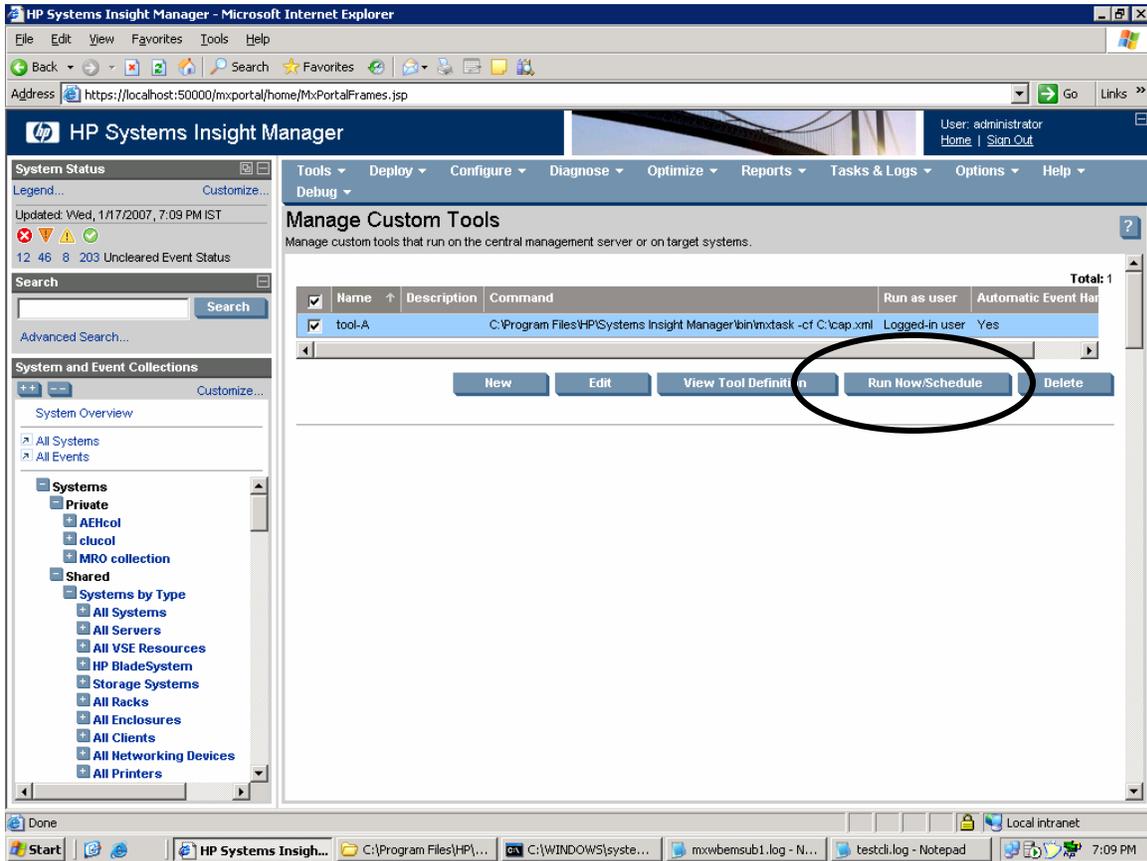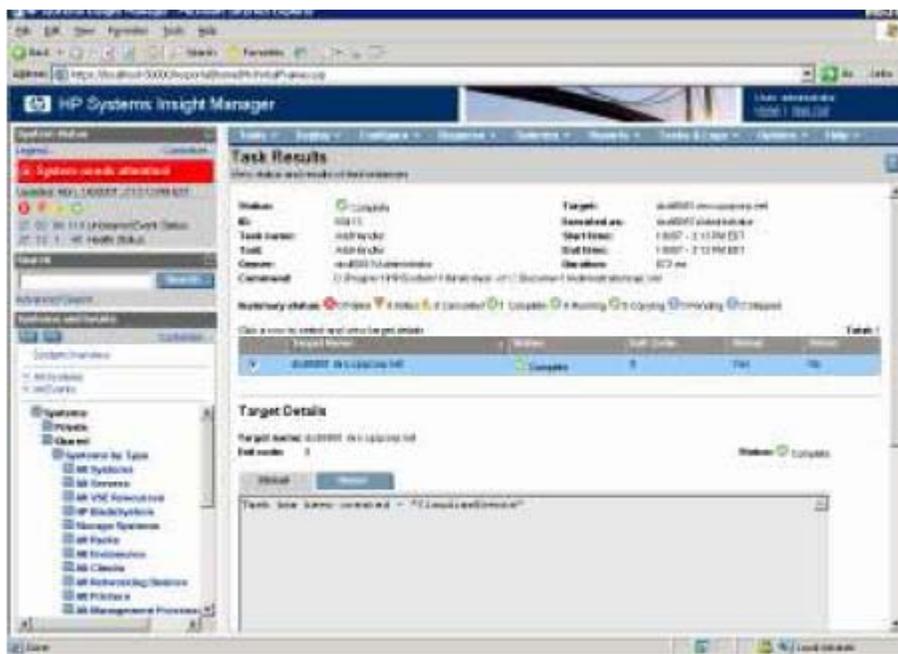5. To test the new tool, select it and click **Run Now/Schedule**.

6. Below is an example of a Tool that was successfully created.

Removing event handler tool

You must create a Remove Event Handler tool that executes when the partner CMS is available again. This disables and removes the event handlers for the other CMS.

1. Create another custom tool definition, in this example, the Command Execution path is the same as add task:

   – `/opt/mx/bin/mxtask` (Linux)

   `C:\Program Files\HP\Systems Insight Manager\bin\mxtask` (Windows)
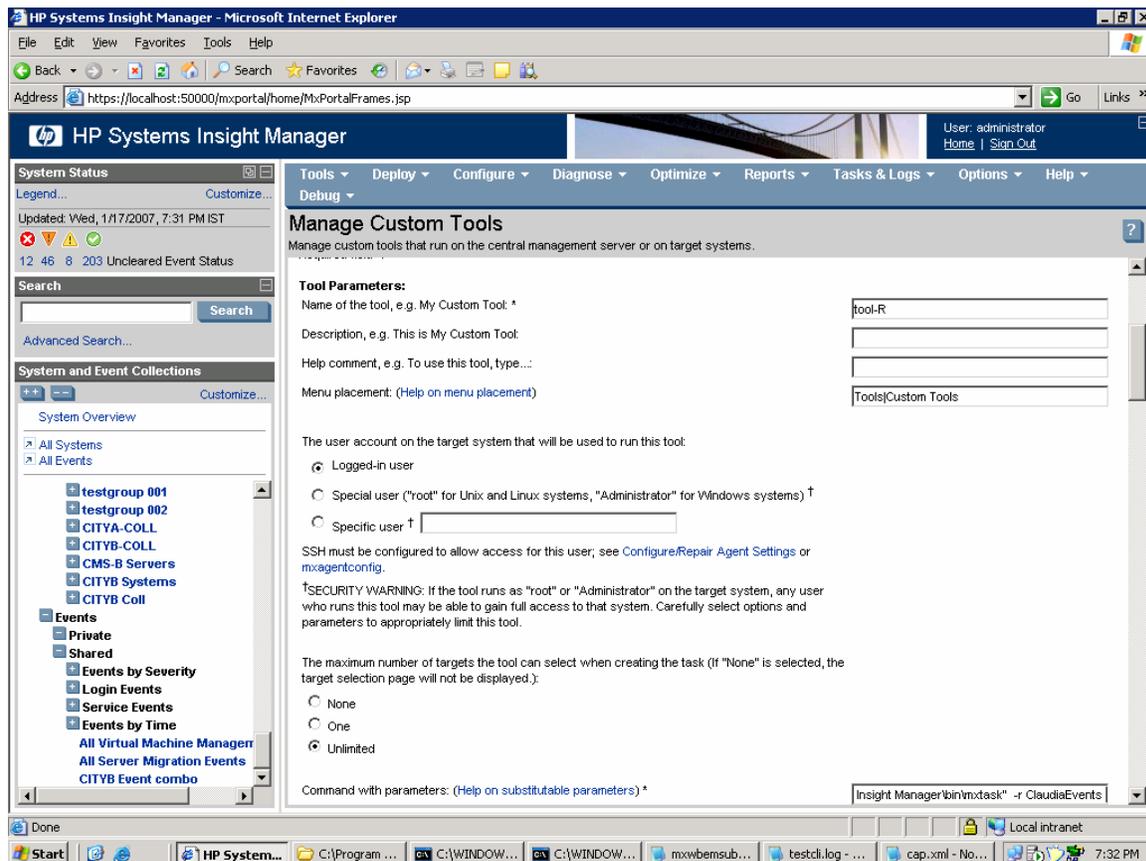
2. Enter the following command to be executed by the tool:

   ```
   C:\Program Files\HP\Systems Insight Manager\bin\mxtask"  -r
   ClaudiaEvents
   ```

3. This time the parameters are `-r ClaudiaEvents` which removes the event handler task, as shown below.

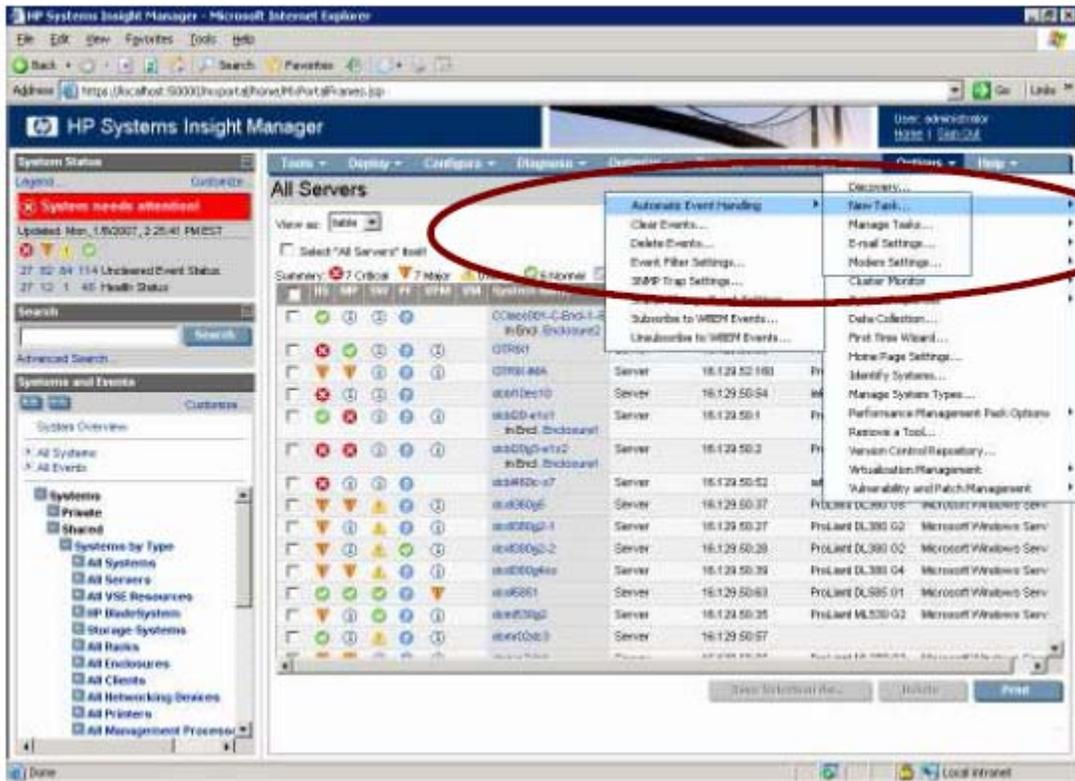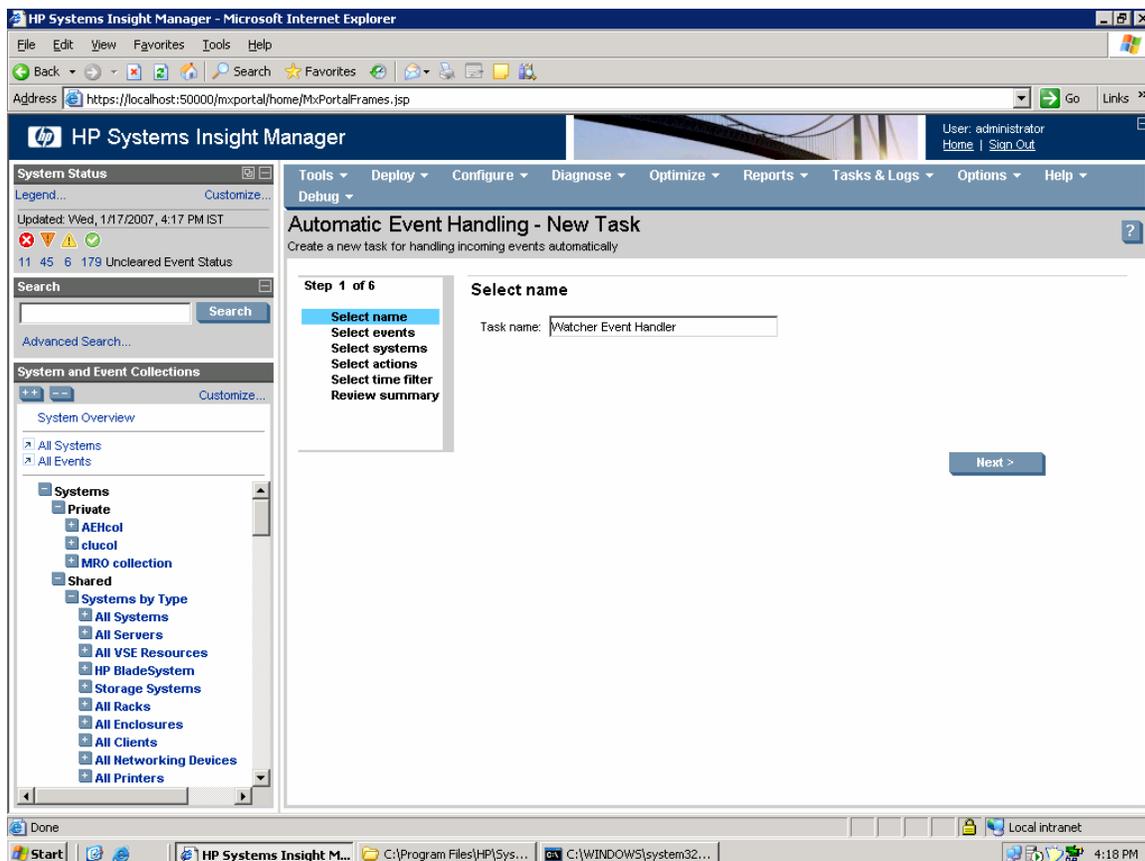## Creating the Watch Dog event handler

Now that we have created our Custom Event System combo collections, one set for City A and another for City B, and we have also created our Add and Remove Event Handlers Custom Tools, we can now create the Event Handler that provides the watch dog service to indicate that the partner CMS has failed and that the AddHandler event needs to created and turned on.

There are the 8 steps to creating an event handler to watch your other HP SIM server, create one on each HP SIM server so that the CMS in City A watch the CMS in City B and vice versa.

From the **Options Menu** select **Events -> Automatic Event Handling -> New Task**:
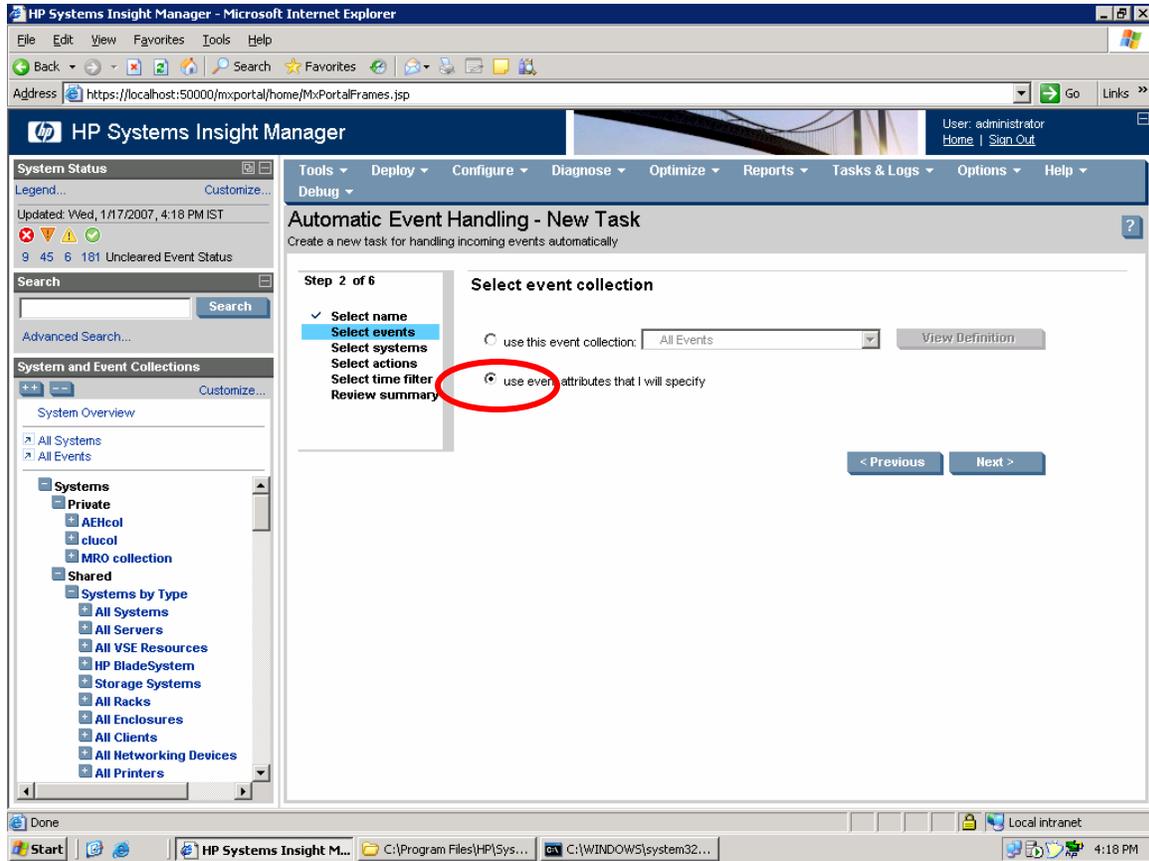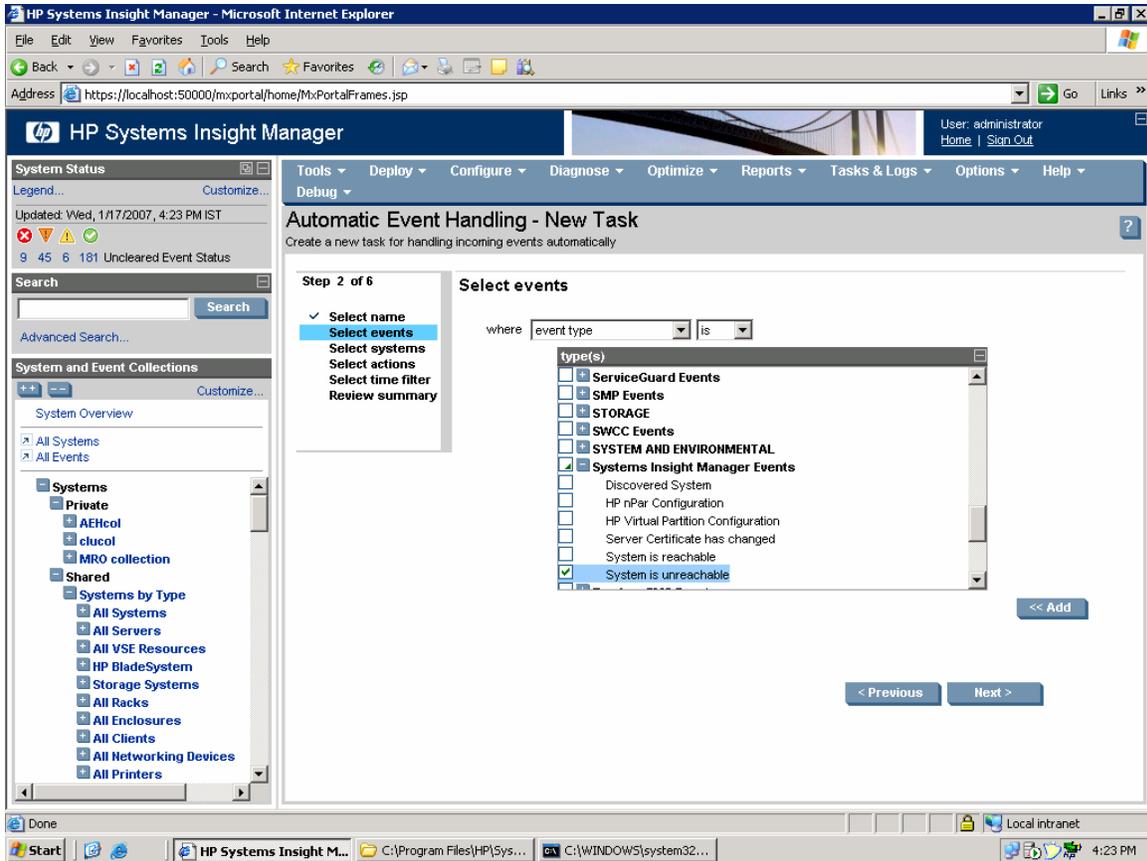
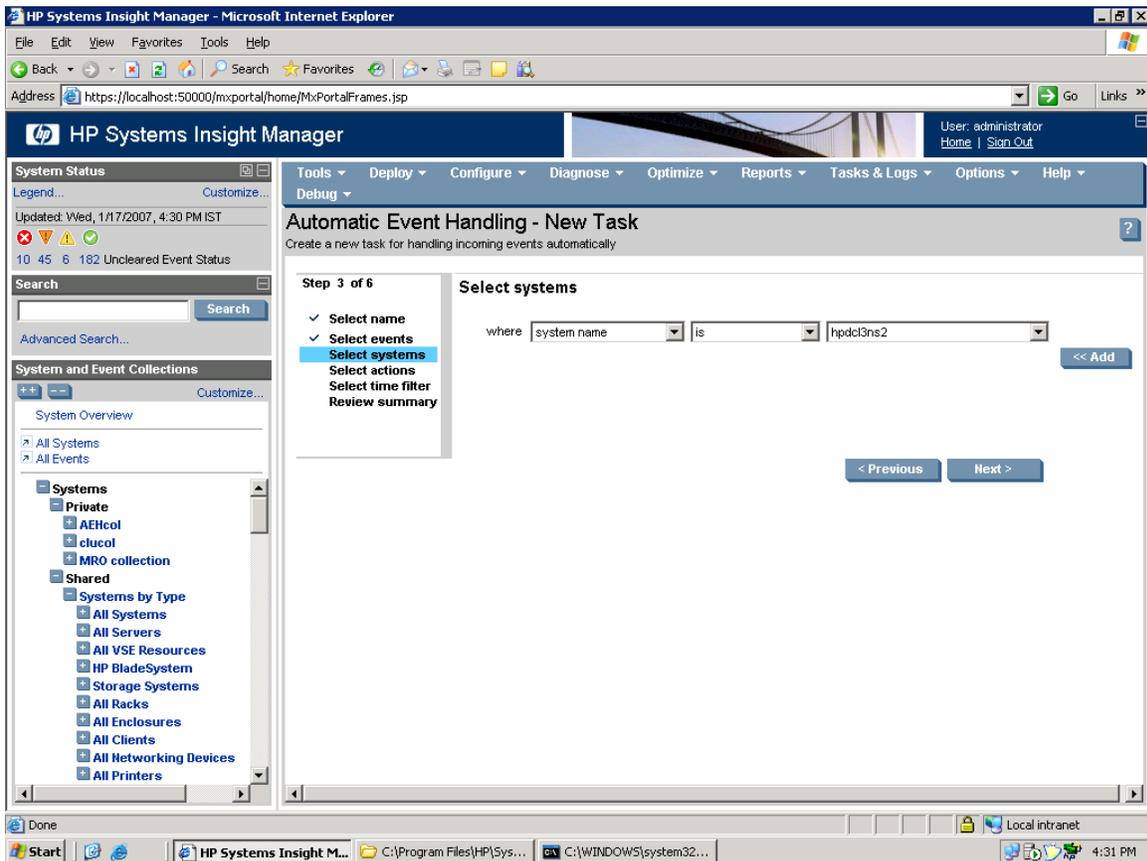1. Give it a name: For this example, name it **Watcher Event Handler**.

2. Select **use event attributes that I will specify** option and then click **Next**.

3. Select the events by **Event Type** where it is a Systems Insight Management Event of **System is unreachable**, then click **Next**.
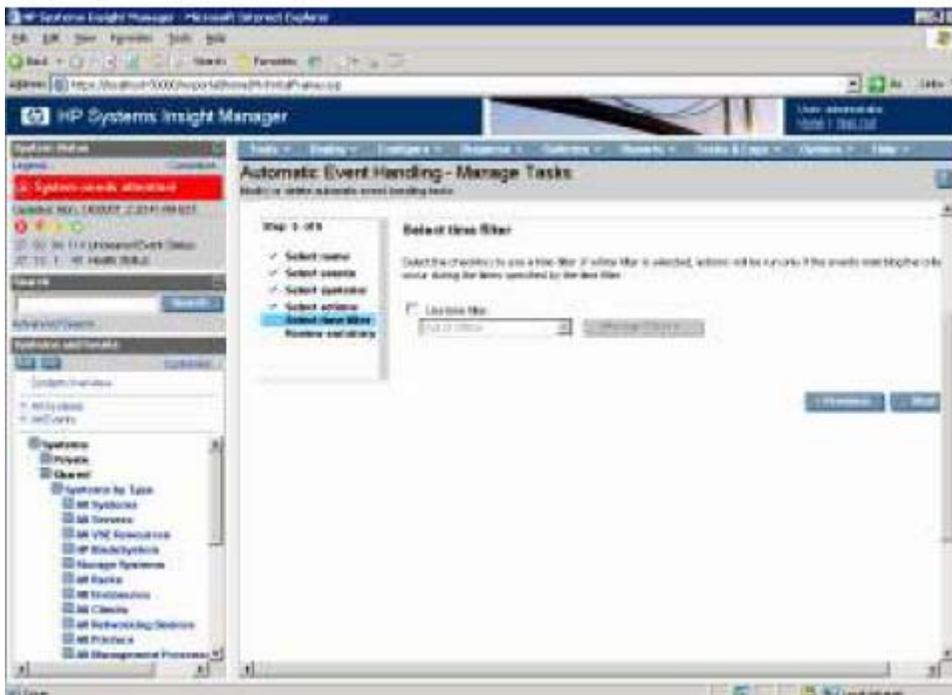
4. Select the other HP SIM server to watch where System Name is the device name of the other HP SIM server.
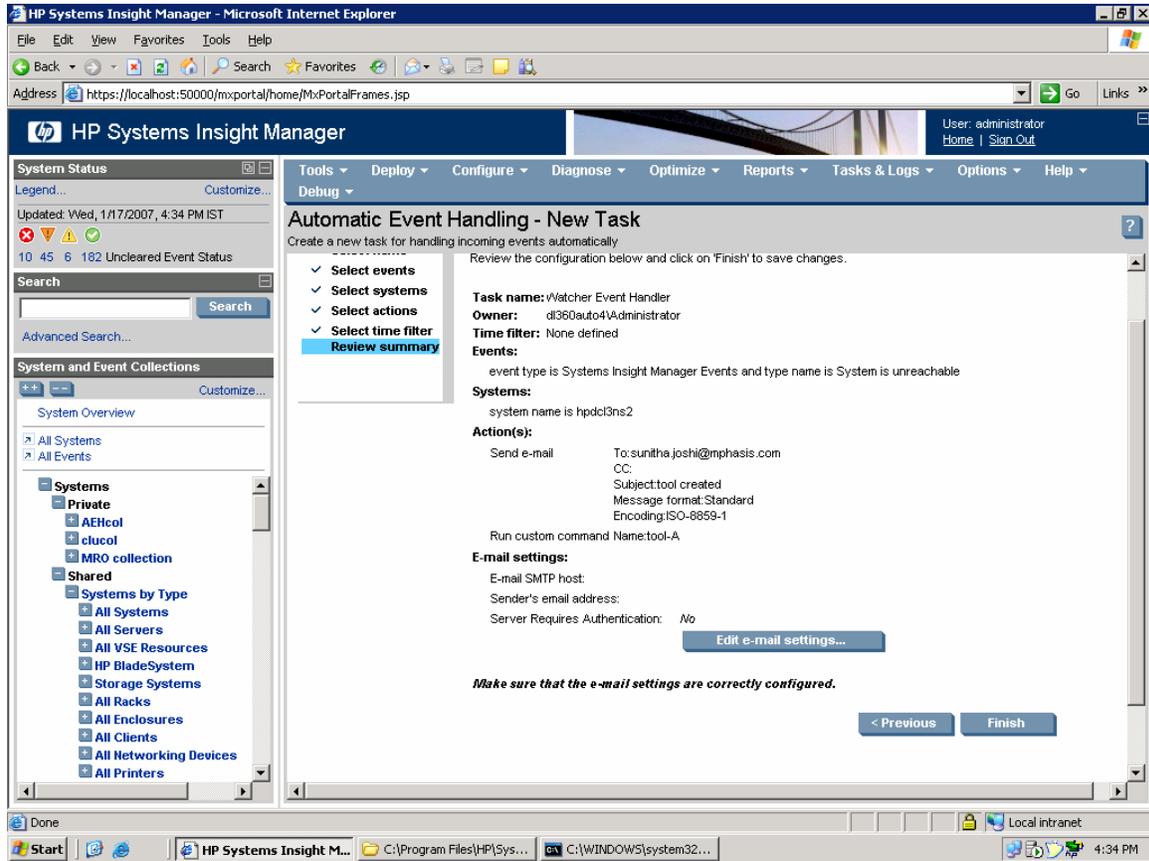
5. Select the Event Tool we created earlier, (in this example, the tool-A Custom Tool) and create an email notification so you know when the event occurs.
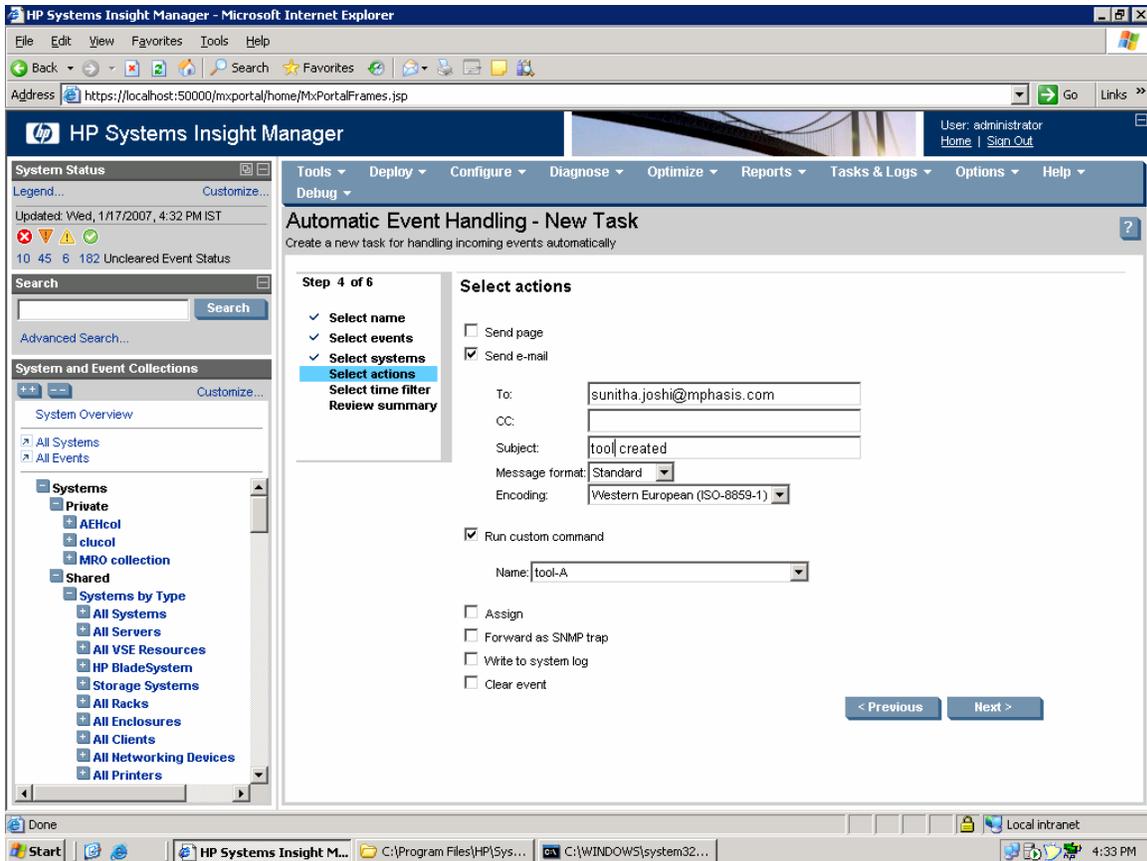
6. No Time Filters are necessary, just click **Next**.

7. Review and finish.

8. Repeat this steps for the Remove Event Handler, by select the status of the other CMS as *Available* and select the Remove Event Handler Custom Tool.

Conclusions

This paper has demonstrated how HP Systems Insight Manager's Command Line Interface can be used to achieve some disaster recovery capabilities. There are many CLI tools that can also be used to automate processes with in HP SIM.

Appendix A – Event handler

Creating an event handler

Using Custom Event System combo Collections created for City A and another for City B, we can now create the Event Handler that is enabled when CITYB CMS is unreachable. This Task is created by the Custom tool.

To get tool definition of this task in a XML format, you must create the event hander shown below and then run the command `mxtask –lf "<Handler name>"`.

There are the 6 steps to creating an event handler which is used to Manage CITYB Systems by CITY A CMS when City B CMS is unreachable. We also create CITYB CMS.
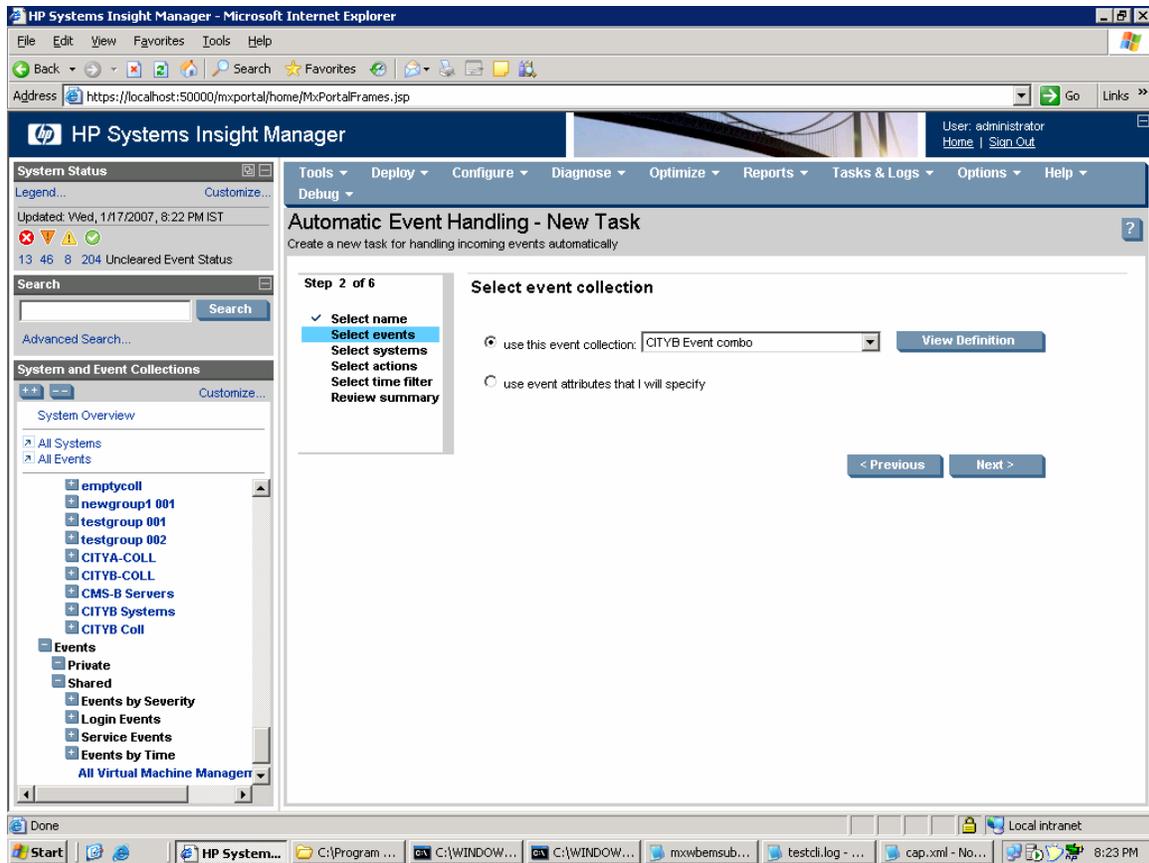
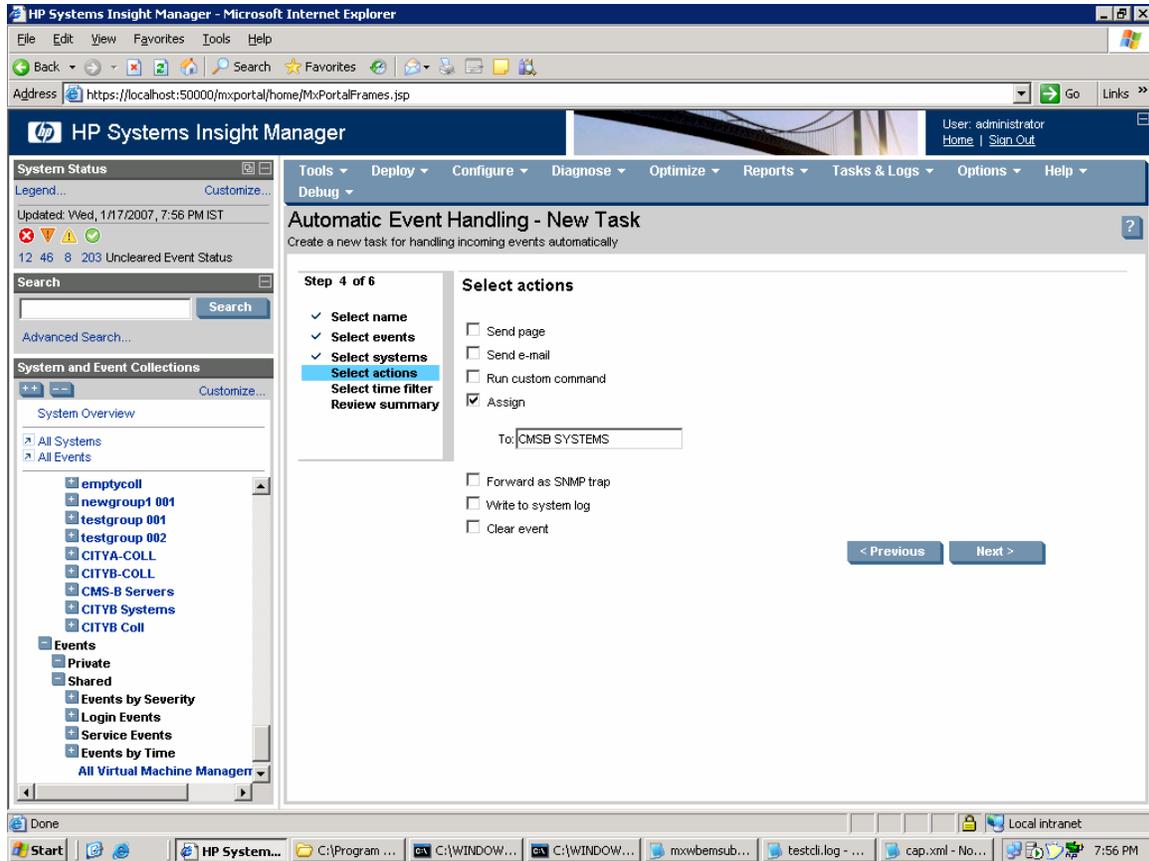From the **Options Menu select Events -> Automatic Event Handling -> New Task**:
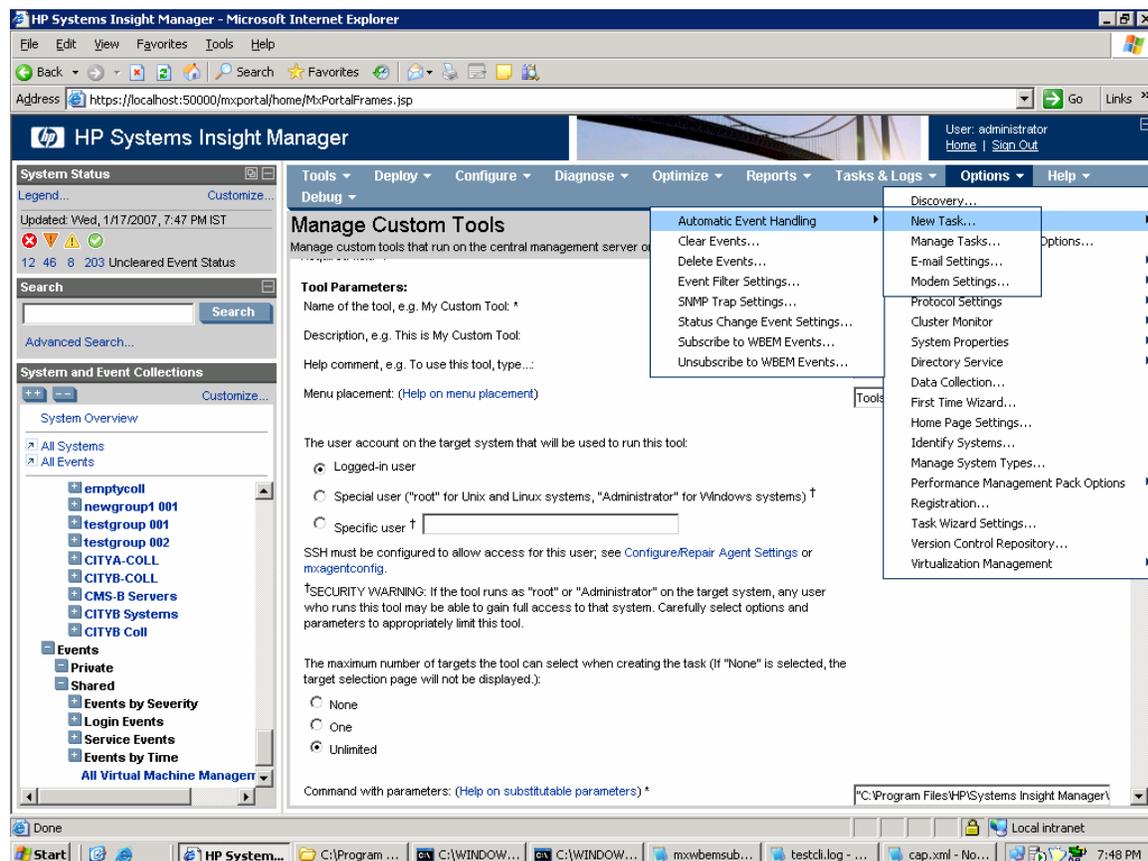
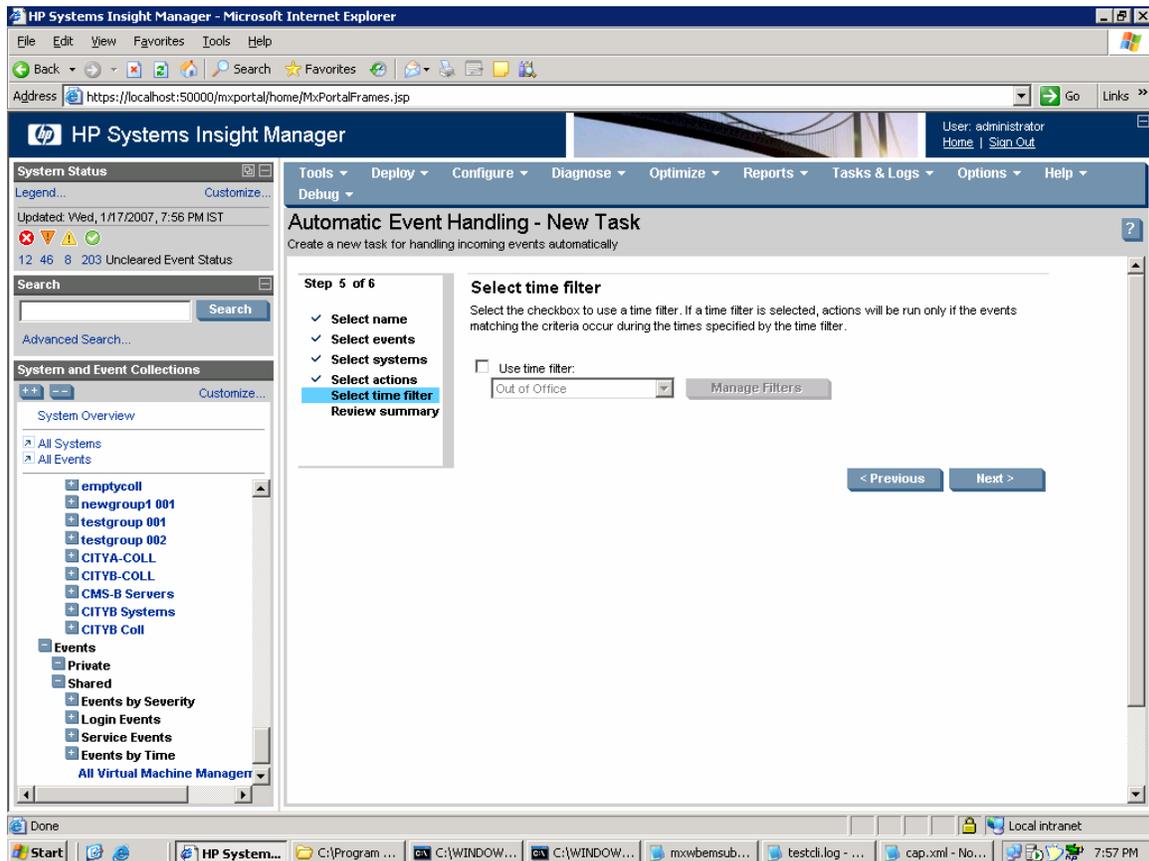1. Enter a name. For this example, use **ClaudiasEvents**.
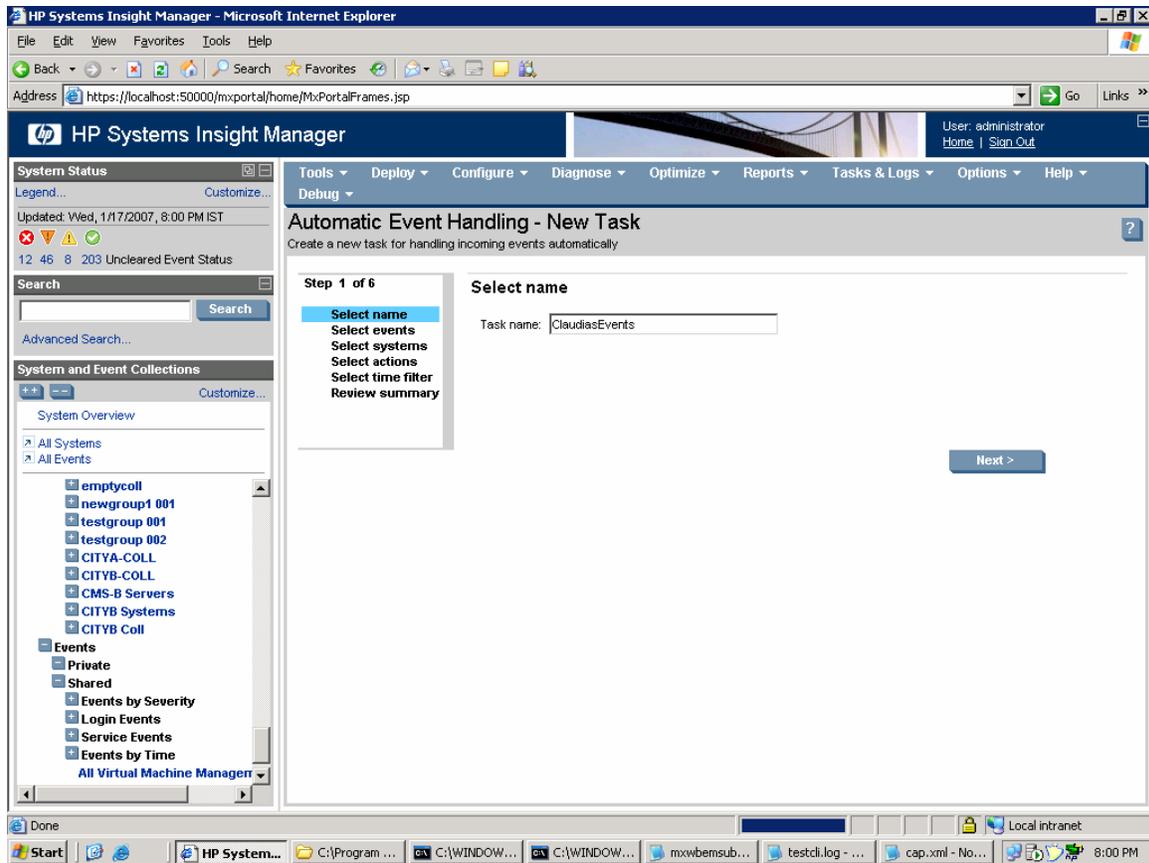
2. Select newly created Event-System combo Collection **CITYB Event combo** collection and then click **Next**.
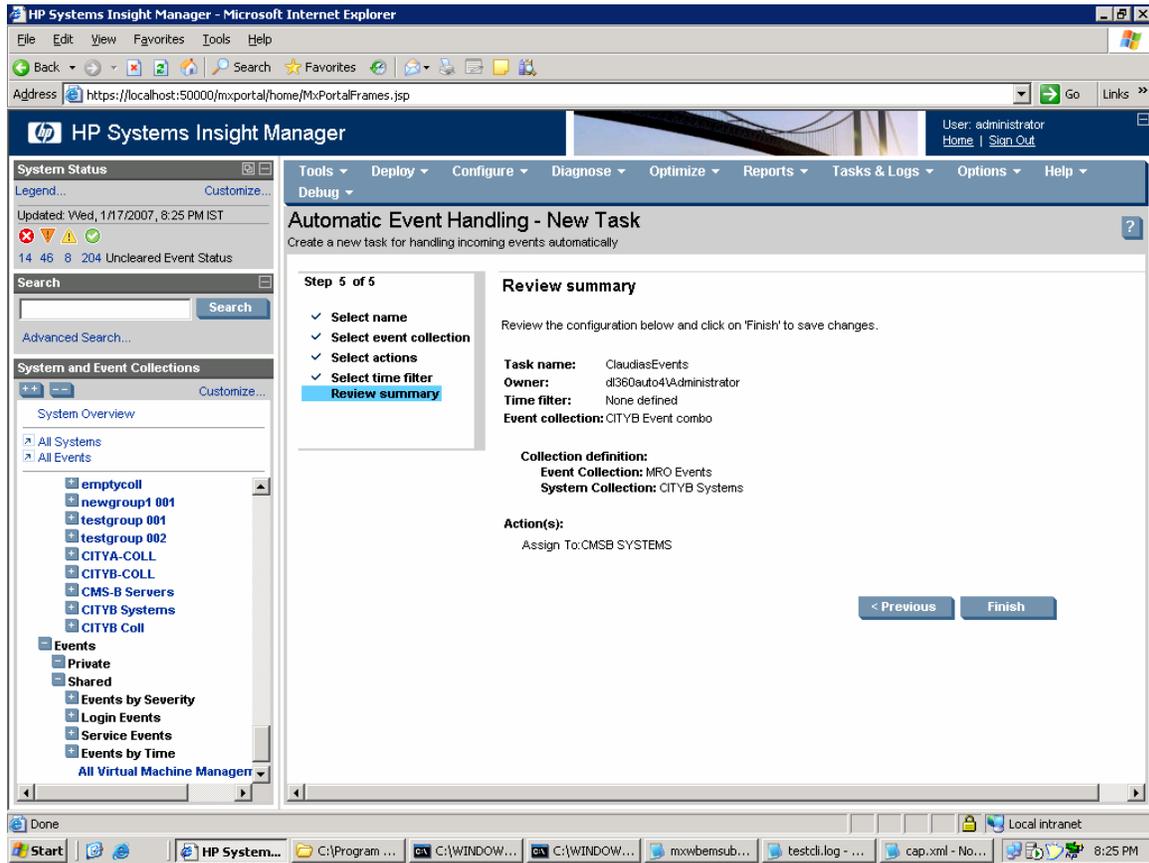
3. Assign it to **CMSB SYSTEMS**, and click **Next**.

4. No Time Filters are necessary. Click **Next**.

5. Review and finish.

6. Repeat this steps for the other CMS.

Appendix B – Sample event handler XML file

Here is the XML extracted using mxtask –lf . Note it uses queryname **CITYB Event combo** .For this example, you can cut & paste from this document. The XML file in the previous example is named *C:\Docume~1\Administrator\cap.xml.* Edit the queryname to match the queries you have created. You can create the Event Handler Task using the GUI (same steps are for creating the Watch Dog Event Handler above); then export it using the CLI tool.

For example; `mxtask –lf ClaudiasEvents >> cap.xml`.

```
<?xml version="1.0" encoding="windows-1252"?>
<task-list>
    <task name="AutomaticEventHandler1" type="add"
owner="dl360auto4\Administrator" state="enabled">
        <toolname>Action on Events</toolname>
        <queryname>CITYB Event combo</queryname>
        <scheduleinfo />
        <timefilter />
        <toolparams>&lt;?xml version="1.0"?&gt;
&lt;XeObject
className="com.hp.mx.core.tools.actiononevents.ActionOnEventsTaskParamete
rs" classVersion="2"&gt;
 &lt;Property name="wizardMode"&gt;
  &lt;Simple&gt;1&lt;/Simple&gt;
 &lt;/Property&gt;
 &lt;Property name="sendPage"&gt;
  &lt;Simple&gt;false&lt;/Simple&gt;
 &lt;/Property&gt;
 &lt;Property name="pagerMode"&gt;
  &lt;Simple&gt;1&lt;/Simple&gt;
 &lt;/Property&gt;
 &lt;Property name="pagerMessage"&gt;
  &lt;Simple&gt;&lt;/Simple&gt;
 &lt;/Property&gt;
 &lt;Property name="pagingUsers"&gt;
  &lt;List size="0"&gt;
  &lt;/List&gt;
 &lt;/Property&gt;
 &lt;Property name="sendMail"&gt;
  &lt;Simple&gt;false&lt;/Simple&gt;
 &lt;/Property&gt;

 &lt;Property name="mailTo"&gt;
  &lt;Simple&gt;&lt;/Simple&gt;
 &lt;/Property&gt;
 &lt;Property name="mailCC"&gt;
  &lt;Simple&gt;&lt;/Simple&gt;
 &lt;/Property&gt;
 &lt;Property name="mailSubject"&gt;
  &lt;Simple&gt;&lt;/Simple&gt;
 &lt;/Property&gt;
 &lt;Property name="mailFormat"&gt;
  &lt;Simple&gt;1&lt;/Simple&gt;
 &lt;/Property&gt;
 &lt;Property name="mailEncoding"&gt;
  &lt;Simple&gt;UTF-8&lt;/Simple&gt;
 &lt;/Property&gt;
 &lt;Property name="launchApp"&gt;
  &lt;Simple&gt;false&lt;/Simple&gt;
```

```
&lt;/Property&gt;
&lt;Property name="appLaunchTool"&gt;
 &lt;Simple&gt;&lt;/Simple&gt;
&lt;/Property&gt;
&lt;Property name="appLaunchTaskName"&gt;
 &lt;Simple&gt;&lt;/Simple&gt;
&lt;/Property&gt;
&lt;Property name="forwardTrap"&gt;
 &lt;Simple&gt;false&lt;/Simple&gt;
&lt;/Property&gt;
&lt;Property name="trapRecepients"&gt;
 &lt;List size="0"&gt;
 &lt;/List&gt;
&lt;/Property&gt;
&lt;Property name="assignEvent"&gt;
 &lt;Simple&gt;true&lt;/Simple&gt;
&lt;/Property&gt;
&lt;Property name="assignee"&gt;
 &lt;Simple&gt;CMS-A&lt;/Simple&gt;
&lt;/Property&gt;
&lt;Property name="clearEvent"&gt;
 &lt;Simple&gt;false&lt;/Simple&gt;
&lt;/Property&gt;
&lt;Property name="logEvent"&gt;
 &lt;Simple&gt;false&lt;/Simple&gt;
&lt;/Property&gt;
&lt;Property name="forwardWbem"&gt;

 &lt;Simple&gt;false&lt;/Simple&gt;
&lt;/Property&gt;
&lt;Property name="WbemURL"&gt;
 &lt;Simple&gt;&lt;/Simple&gt;
&lt;/Property&gt;
&lt;/XeObject&gt;
</toolparams>
    </task>
</task-list>
```
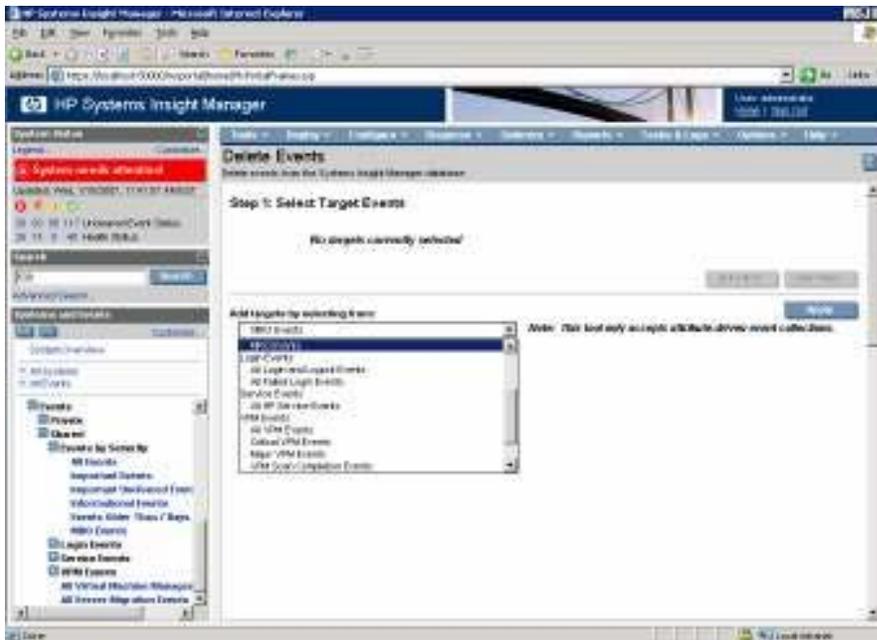
Appendix B - Purging unwanted events

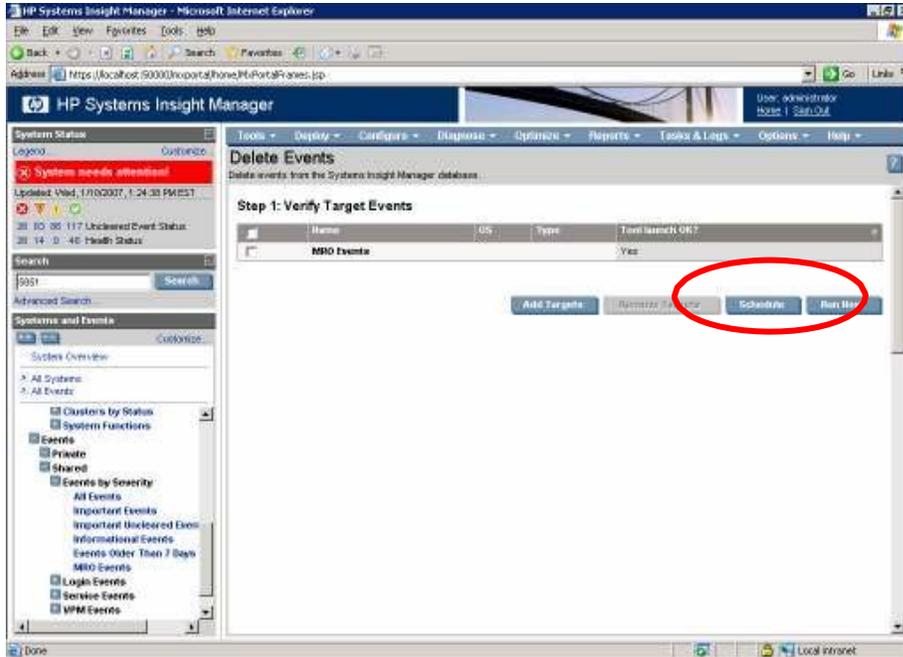To purge unwanted or old events from HP SIM:

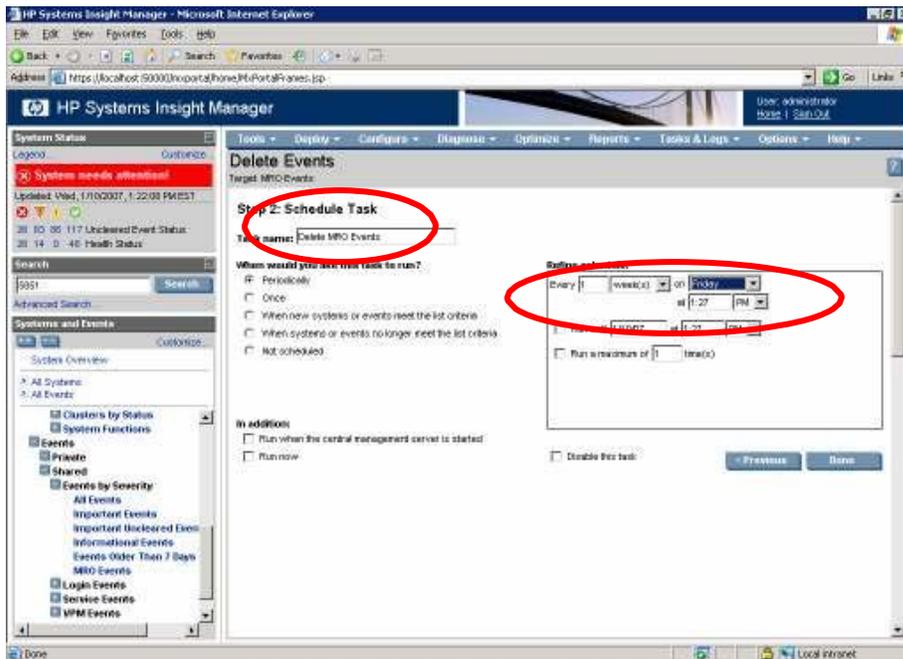1. Click **Options -> Events -> Delete Events** menu selection.
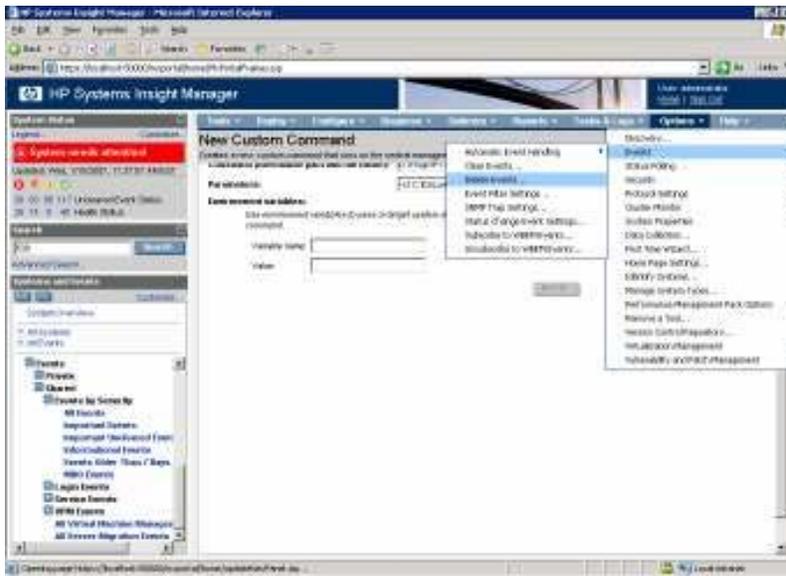
2. Select the Event Collection. For this example, use **MRO Events**.

3. Click **Apply** and click **Schedule**.

4. Enter a name for the task. For this example, **Delete MRO Events** and Schedule it to run once a week. This is a good idea to do for all events on a weekly or monthly basis as it helps keep the CMS database cleaner and more responsive. This allows you to keep a back log of a weeks worth of events.

For more information