Xtremesoft

Using AppMetrics to Handle Hung Components and Applications

AppMetrics has the ability to detect hung COM+ components and applications, providing a means of alerting staff to situations where critical applications are no longer running properly. A key feature of AppMetrics is its ability to trigger a restart of those hung components and applications.

This document describes the steps necessary to setup AppMetrics to detect hung components and deliver notifications. For information on how to setup AppMetrics in order to restart hung components, please refer to the section ***Using AppMetrics to Restart Hung Components*** at the end of this document.

Table of Contents

Setting up AppMetrics to Detect Hung Components

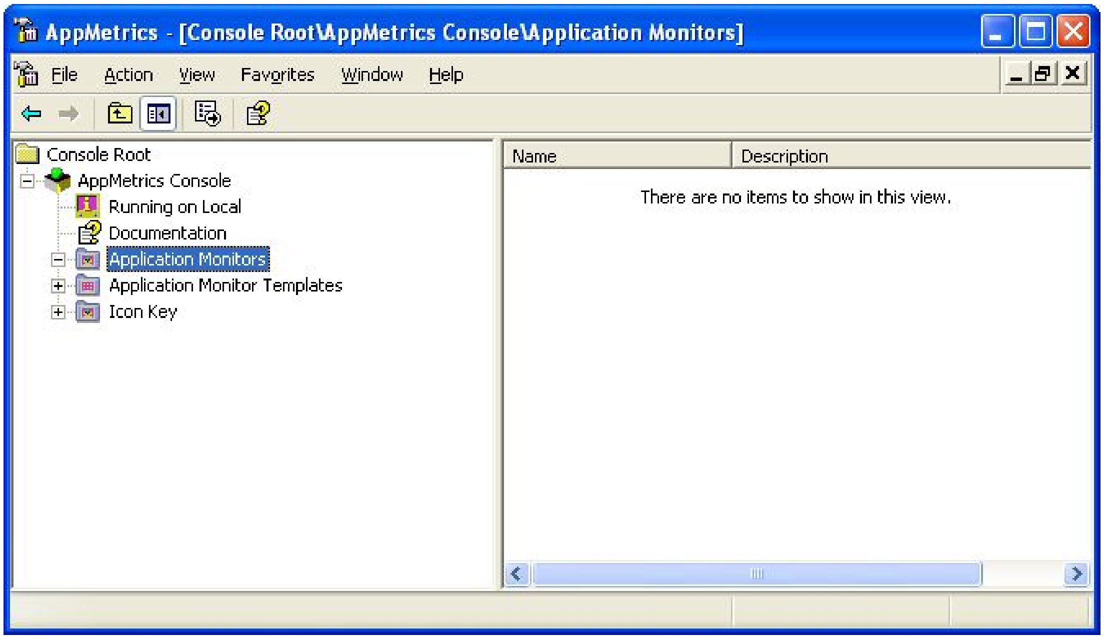
In order to use AppMetrics to detect hung components, a Production Monitor must be created and configured in order to monitor the desired group of applications. Notifications will also need to be configured in order for alerts to be delivered by email, logged to the Windows Event Log, or for them to be managed by SNMP. Additionally, a special windows script, XSNotification.wsc, can optionally be installed and configured in order to demonstrate the functionality that a custom script provides in relation to AppMetrics alerts.

Chapter 1. Creating the Monitor

In order for alerts to be generated a AppMetrics Production Monitor must be used. The following procedure describes how to create and configure a Production Monitor and its associated Agent.

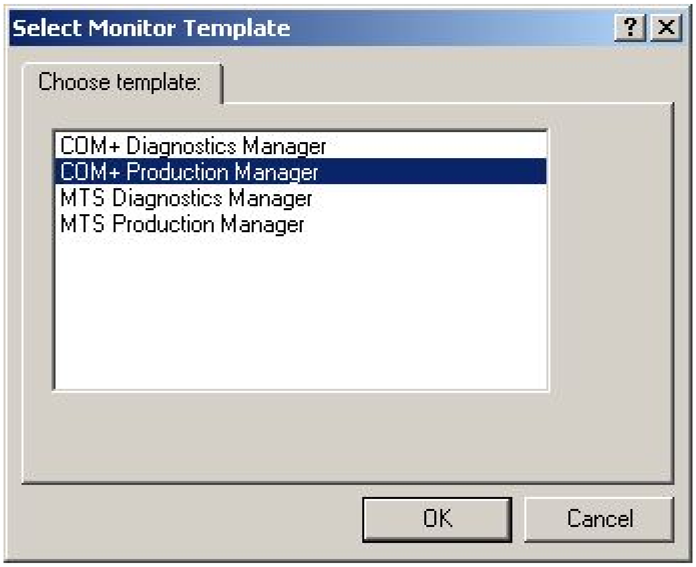
Creating a New Production Monitor

1. From the ***AppMetrics console***, right click on **Application Monitors**, and from the menu select **New->Application Monitor**.

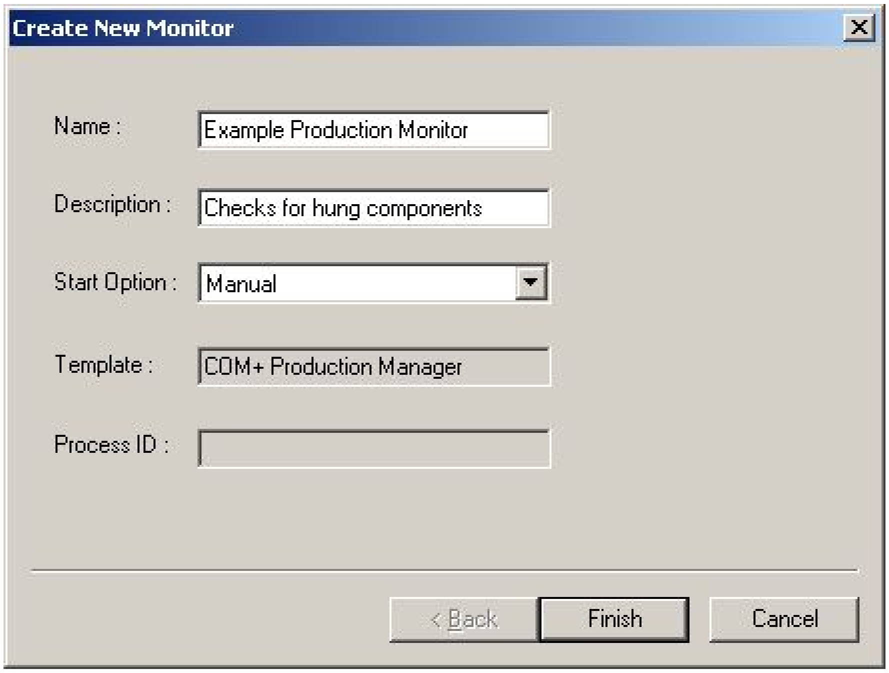


There are no items to show in this view.

2. Select the **COM+ Production Manager** item from the ***Select Monitor Template*** dialog box.



3. In the ***Create New Monitor*** dialog box, enter a name for the monitor. An optional description may also be entered if so desired.



Name : Example Production Monitor

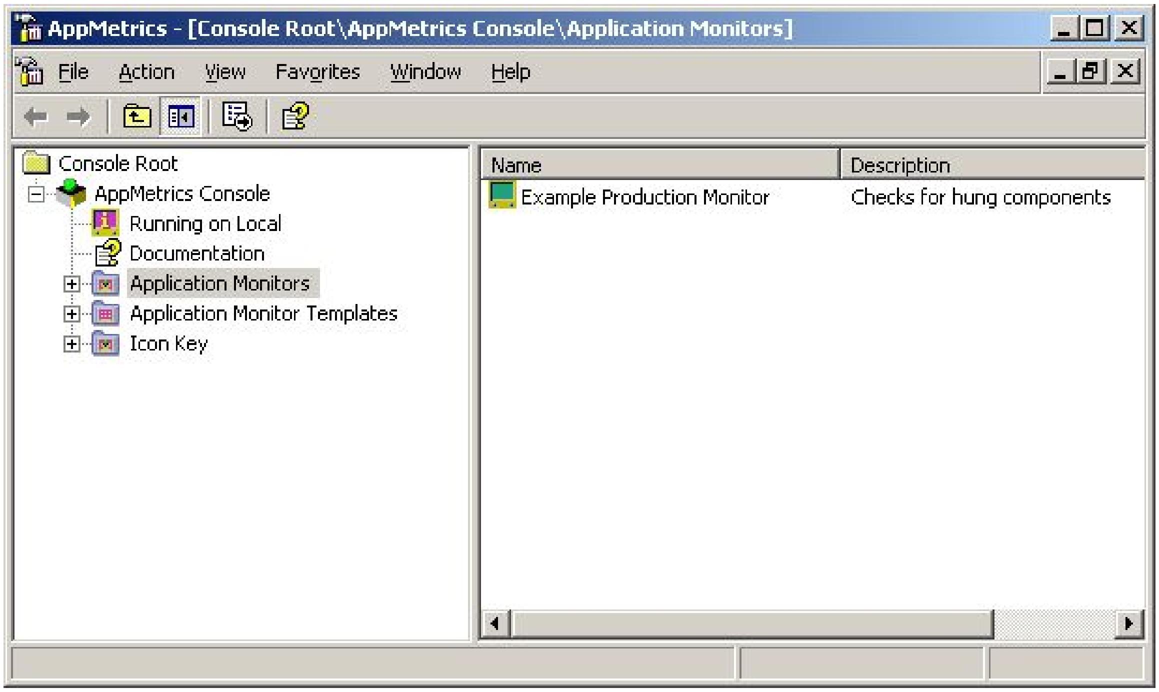
Description : Checks for hung components

Start Option : Manual

Template : COM+ Production Manager

Process ID :

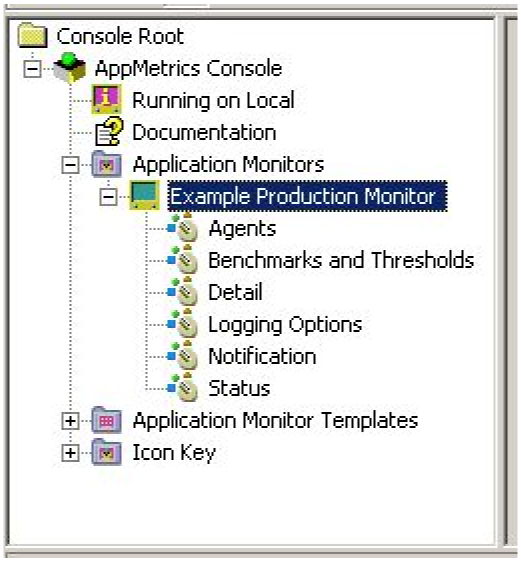
4. Click **Finish**. The new monitor will now be created. Once complete, the console should display the monitor in the right hand pane as shown below.



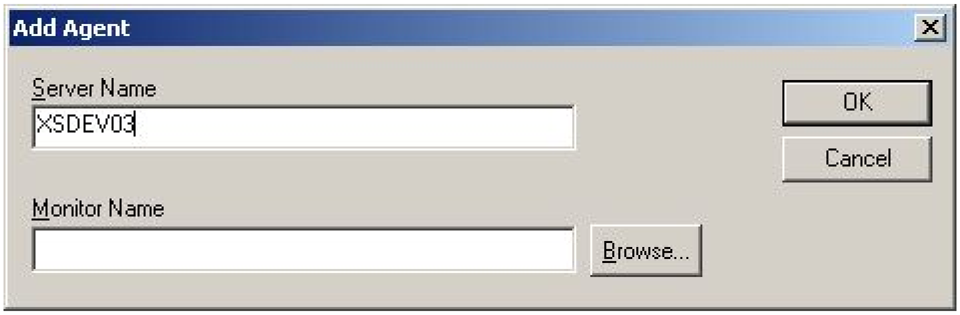
Name	Description
Example Production Monitor	Checks for hung components

Creating a New Production Agent for the Monitor

1. In the ***AppMetrics console*** left hand pane, expand the **Application Monitors** node, then expand the monitor node (in our example we use the name Example Production Monitor).



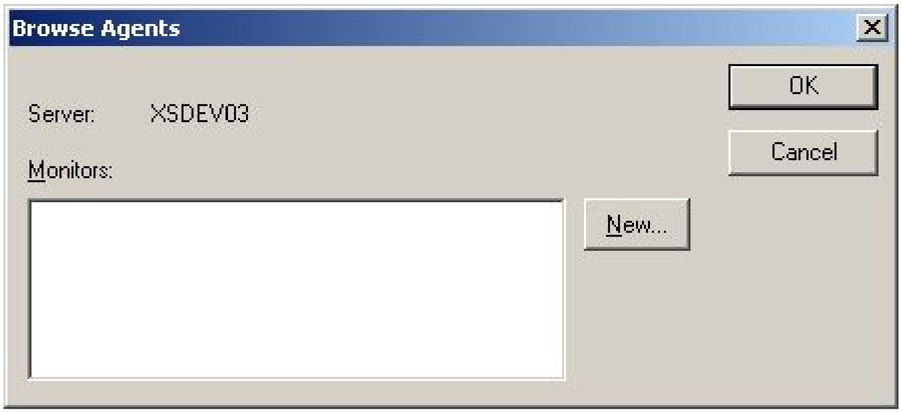
2. Right click the **Agent** node, and select **New->Agent** from the menu.



Server Name: XSDEV03

Monitor Name:

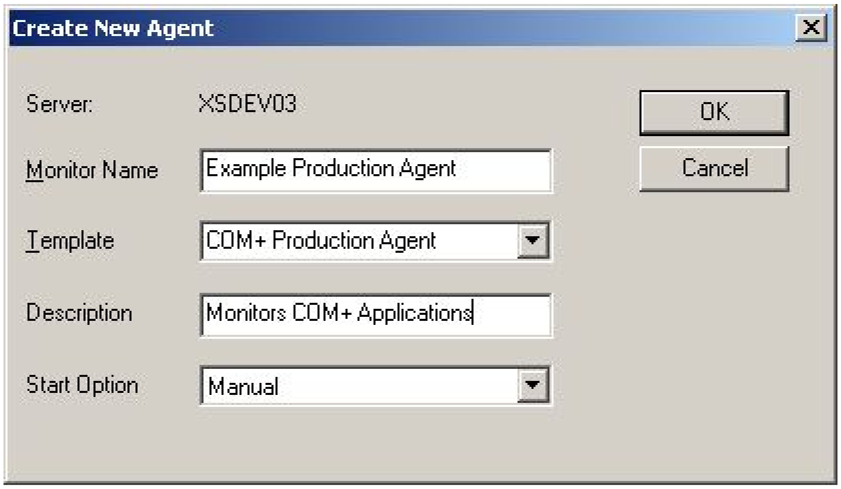
3. In the ***Add Agent*** dialog box, enter the name of the server that you wish to monitor, and then select the **Browse** button.



Server: XSDEV03

Monitors:

4. Select the **New** button on the ***Browse Agents*** dialog box.



Server: XSDEV03

Monitor Name: Example Production Agent

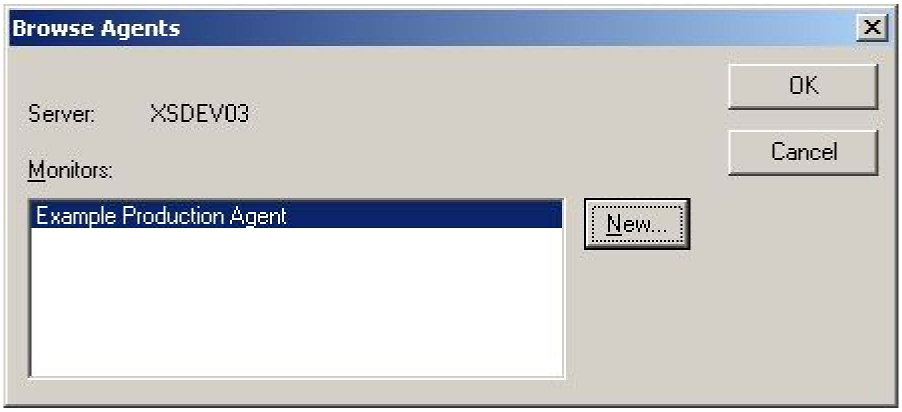
Template: COM+ Production Agent

Description: Monitors COM+ Applications

Start Option: Manual

5. In the ***Create New Agent*** dialog box, enter a name for the Agent. An optional description may be entered if desired.

6. Select **OK** on the ***Create New Agent*** dialog box.

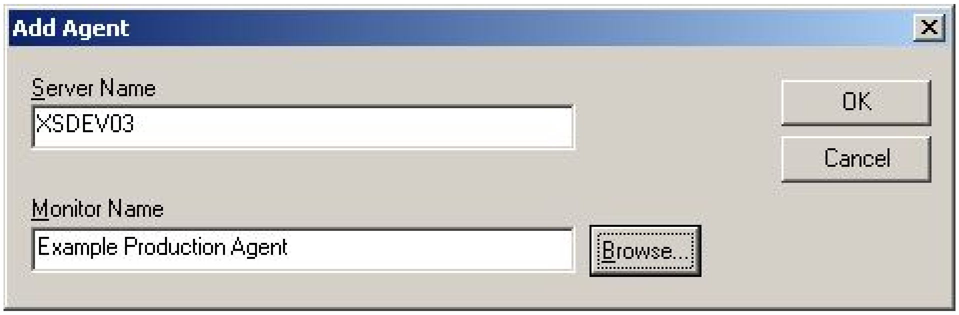


Server: XSDEV03

Monitors:

Example Production Agent

7. Select **OK** on the ***Browse Agents*** dialog box.

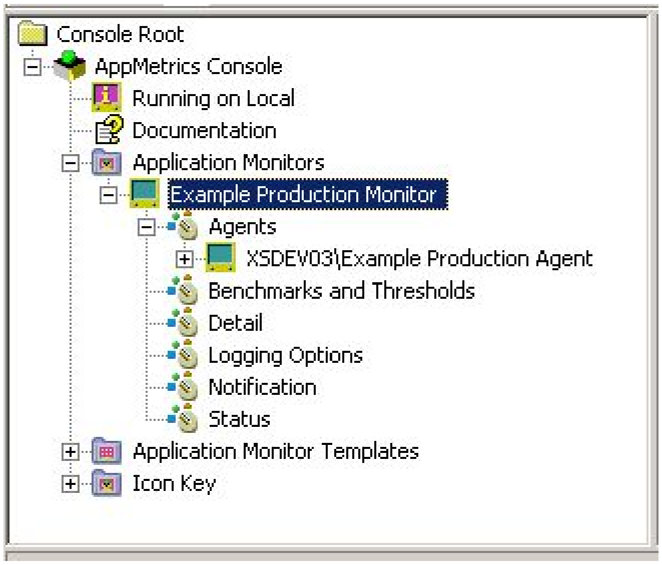


Server Name: XSDEV03

Monitor Name: Example Production Agent

8. Select **OK** on the ***Add Agent*** dialog box.

9. The ***AppMetrics console*** left hand pane should appear as shown below



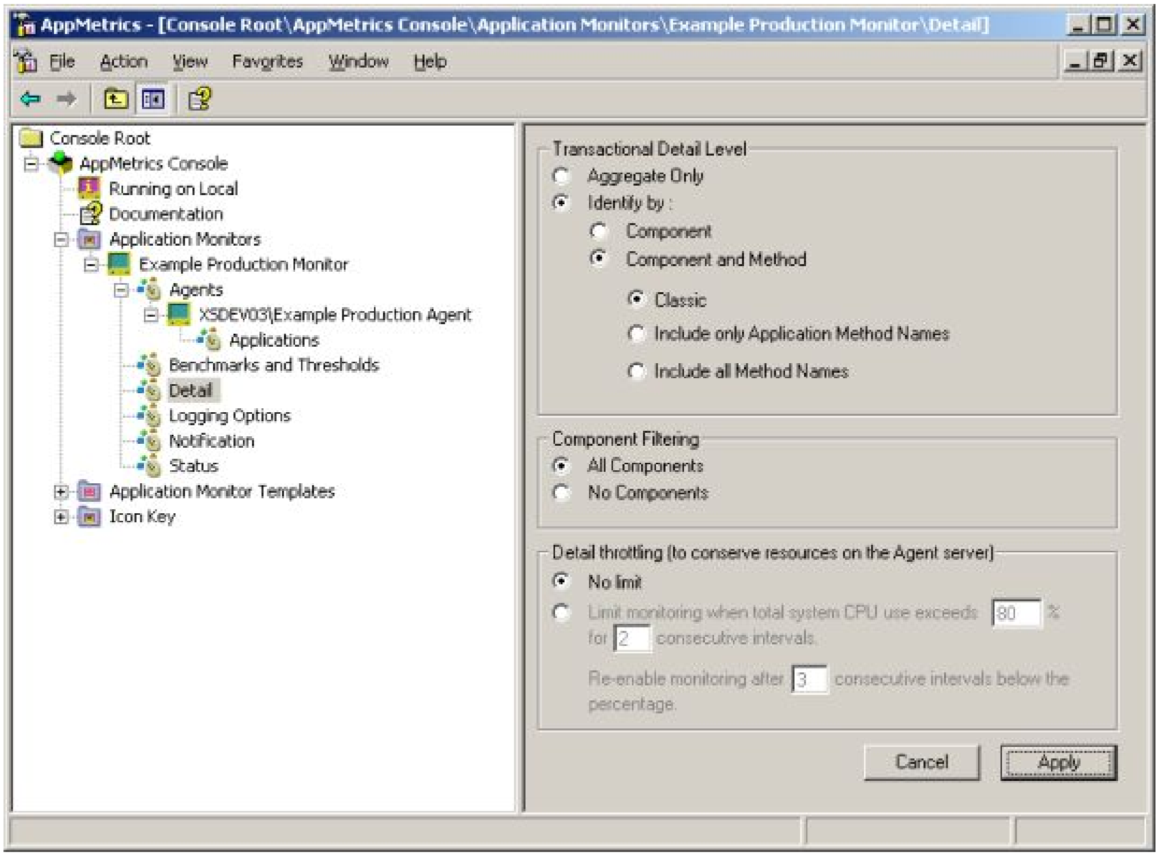
- Console Root
  - AppMetrics Console
    - Running on Local
    - Documentation
    - Application Monitors
      - Example Production Monitor
        - Agents
          - XSDEV03\Example Production Agent
        - Benchmarks and Thresholds
        - Detail
        - Logging Options
        - Notification
        - Status
    - Application Monitor Templates
    - Icon Key

Chapter 2. Configuring the Monitor

Here we will configure the manager and the agent to monitor our desired applications.

Configuring the Manager

1. In the left pane of the ***AppMetrics console***, select the Production Monitor **Detail** node.



AppMetrics - [Console Root\AppMetrics Console\Application Monitors\Example Production Monitor\Detail]

Transactional Detail Level

○ Aggregate Only
● Identify by :
　○ Component
　● Component and Method
　　● Classic
　　○ Include only Application Method Names
　　○ Include all Method Names

Component Filtering

● All Components
○ No Components

Detail throttling (to conserve resources on the Agent server)

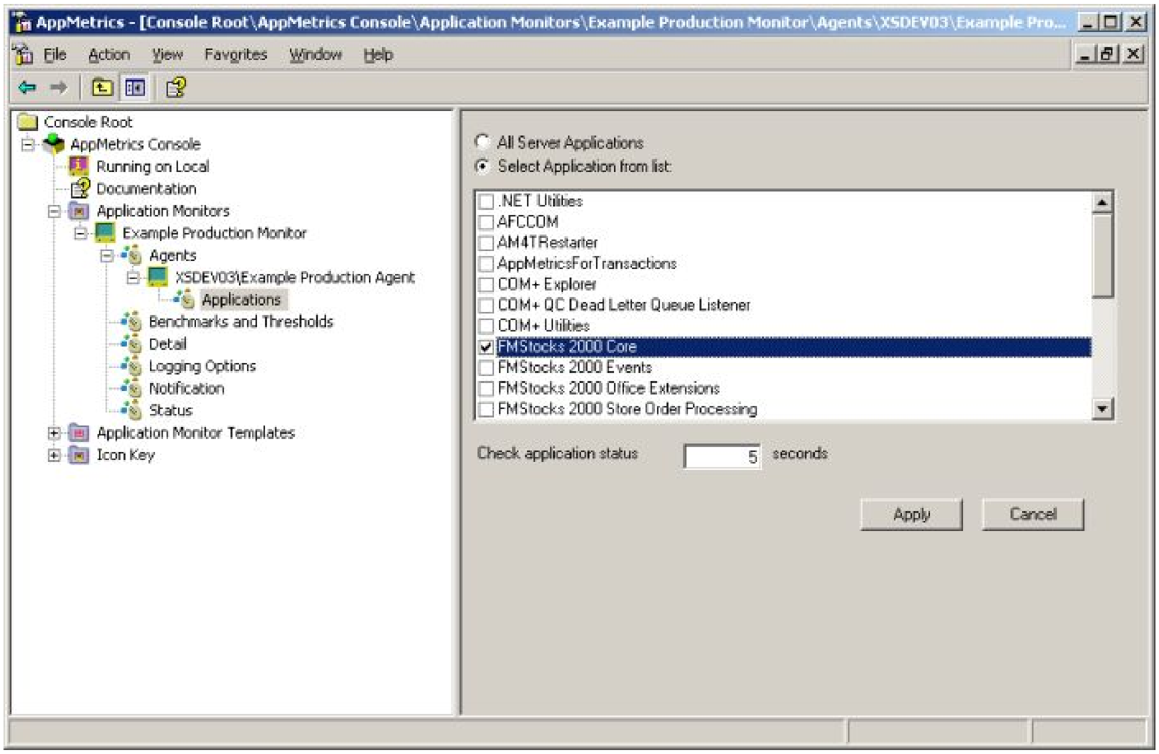
● No limit
○ Limit monitoring when total system CPU use exceeds 80 % for 2 consecutive intervals.
Re-enable monitoring after 3 consecutive intervals below the percentage.

2. The **Transactional Detail Level** should be modified to **Identify by** either **Component** or **Component and Method**. This is necessary in order to later set the Component Threshold.

3. Select the **Apply** button to save the new settings.

Configuring the Agent

1. In the left pane of the ***AppMetrics console***, select the Agent **Applications** node.


AppMetrics - [Console Root\AppMetrics Console\Application Monitors\Example Production Monitor\Agents\XSDEV03\Example Pro...
File
Action
View
Favorites
Window
Help
Console Root
AppMetrics Console
Running on Local
Documentation
Application Monitors
Example Production Monitor
Agents
XSDEV03\Example Production Agent
Applications
Benchmarks and Thresholds
Detail
Logging Options
Notification
Status
Application Monitor Templates
Icon Key
Apply
Cancel

○ All Server Applications
● Select Application from list:

□ .NET Utilities
□ AFCCOM
□ AM4TRestarter
□ AppMetricsForTransactions
□ COM+ Explorer
□ COM+ QC Dead Letter Queue Listener
□ COM+ Utilities
☑ FMStocks 2000 Core
□ FMStocks 2000 Events
□ FMStocks 2000 Office Extensions
□ FMStocks 2000 Store Order Processing

Check application status 5 seconds

2. You may choose to monitor all COM+ applications on the server, or pick one or more specific applications to monitor. To monitor individual applications, select the **Select Application from list** button to choose individual applications from the list, and then select the desired application. The **All Server Applications** selection will cause the Agent to monitor all applications on the server. This is the default setting.

3. Select the **Apply** button to save the new settings.

Chapter 3. Setting the Application Benchmarks and Thresholds

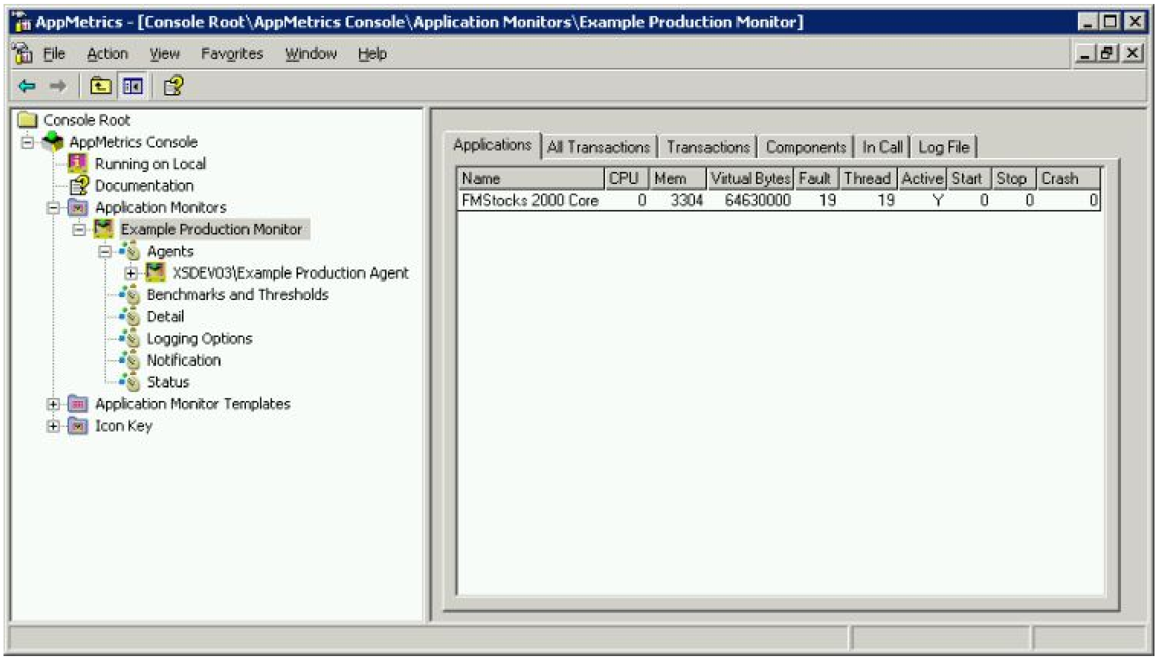
It is necessary to configure the application benchmarks and thresholds in order for AppMetrics to detect a hung component or application. This section describes the steps required to properly configure AppMetrics in order to properly set the necessary benchmarks and thresholds.

Collecting Application Benchmark Data

In order to provide a benchmark for thresholds to be set, it is necessary to collect data for a period of time where a known amount of traffic is being generated by your application. This involves running your application at a time where there is either a known load or at a time of the day where normal traffic is applied to the application.

1. Ensure that your application is running under load.

2. Start the AppMetrics Production Monitor by right clicking on the monitor node in the left hand panel of the ***AppMetrics console***, and select **Start Monitor**.



Name	CPU	Mem	Virtual Bytes	Fault	Thread	Active	Start	Stop	Crash
FMStocks 2000 Core	0	3304	64630000	19	19	Y	0	0	0

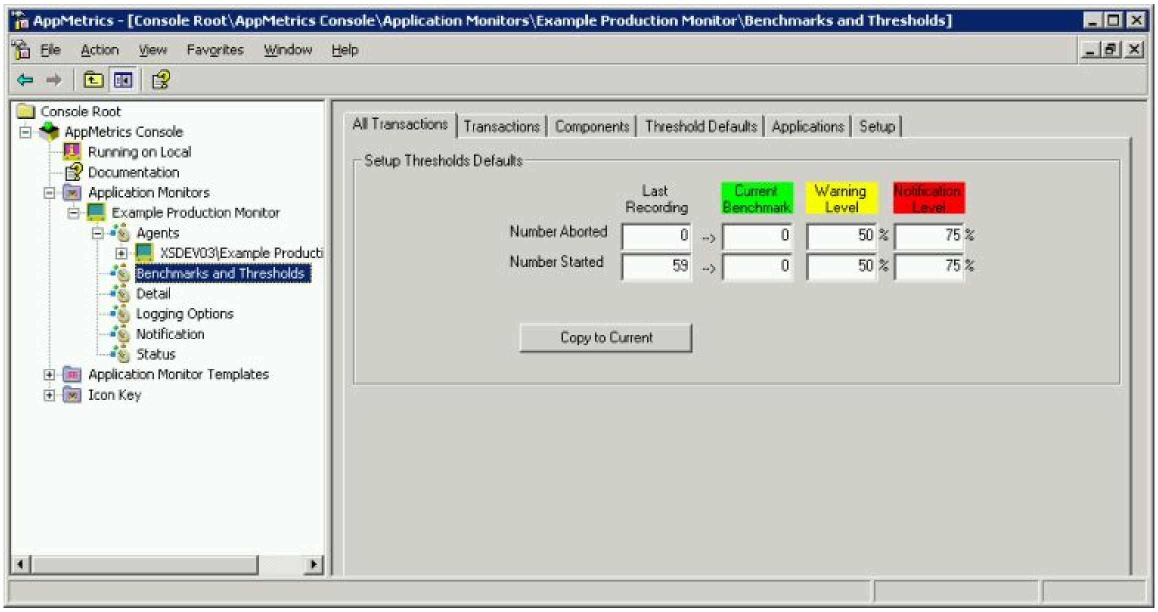
3. Allow AppMetrics to collect data for a period of time where a reasonable sampling of the traffic has taken place, typically about an hour.

4. Right click on the monitor node in the left hand panel of the ***AppMetrics console*** and select **Stop Monitor**.

Configuring Benchmarks and Thresholds

Now that benchmark data has been collected, it is possible to configure the application benchmark and thresholds settings

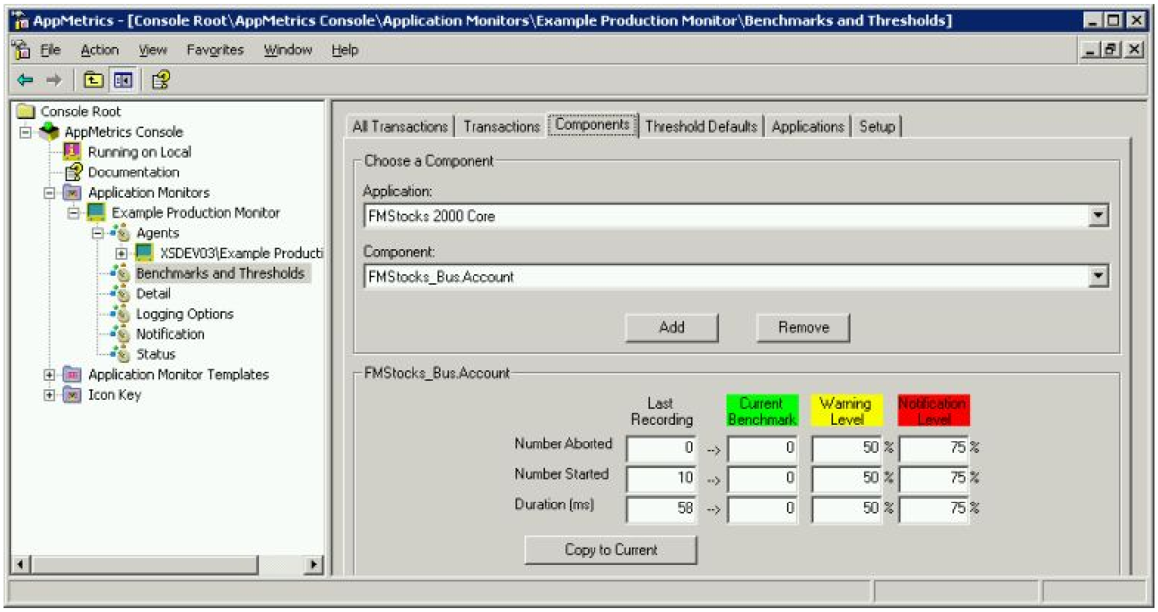
1. Select the **Benchmarks and Thresholds** node from the ***AppMetrics console***.


AppMetrics - [Console Root\AppMetrics Console\Application Monitors\Example Production Monitor\Benchmarks and Thresholds]
File Action View Favorites Window Help
Console Root
AppMetrics Console
Running on Local
Documentation
Application Monitors
Example Production Monitor
Agents
XSDEV03\Example Producti
Benchmarks and Thresholds
Detail
Logging Options
Notification
Status
Application Monitor Templates
Icon Key
All Transactions | Transactions | Components | Threshold Defaults | Applications | Setup
Copy to Current

Setup Thresholds Defaults

	Last Recording		Current Benchmark	Warning Level	Notification Level
Number Aborted	0	-->	0	50 %	75 %
Number Started	59	-->	0	50 %	75 %

2. Select the **Components** page.


AppMetrics - [Console Root\AppMetrics Console\Application Monitors\Example Production Monitor\Benchmarks and Thresholds]
File Action View Favorites Window Help
Console Root
AppMetrics Console
Running on Local
Documentation
Application Monitors
Example Production Monitor
Agents
XSDEV03\Example Producti
Benchmarks and Thresholds
Detail
Logging Options
Notification
Status
Application Monitor Templates
Icon Key
All Transactions | Transactions | Components | Threshold Defaults | Applications | Setup
Add
Remove
Copy to Current

Choose a Component

Application: FMStocks 2000 Core

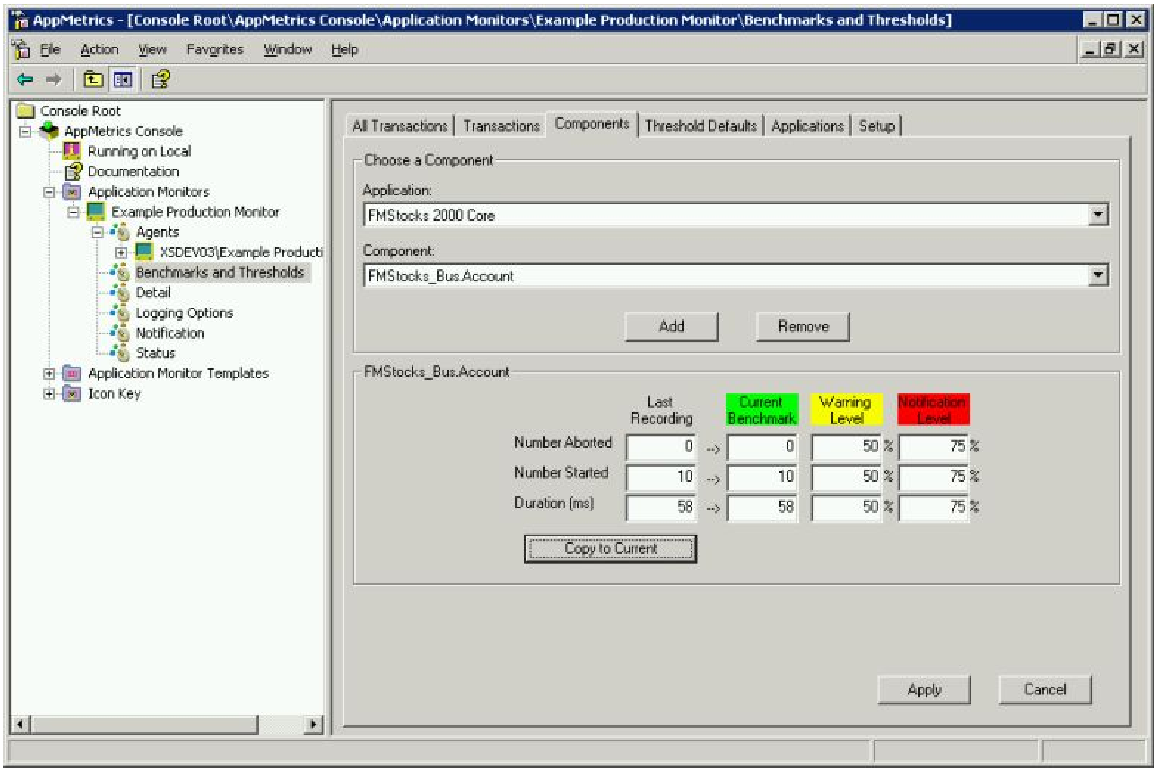
Component: FMStocks_Bus.Account

FMStocks_Bus.Account

	Last Recording		Current Benchmark	Warning Level	Notification Level
Number Aborted	0	-->	0	50 %	75 %
Number Started	10	-->	0	50 %	75 %
Duration (ms)	58	-->	0	50 %	75 %

3. From the ***Component*** page, select the application and component you wish to monitor.

4. Click the **Copy to Current** button. This will save the current values as the benchmark.



AppMetrics - [Console Root\AppMetrics Console\Application Monitors\Example Production Monitor\Benchmarks and Thresholds]

Application: FMStocks 2000 Core

Component: FMStocks_Bus.Account

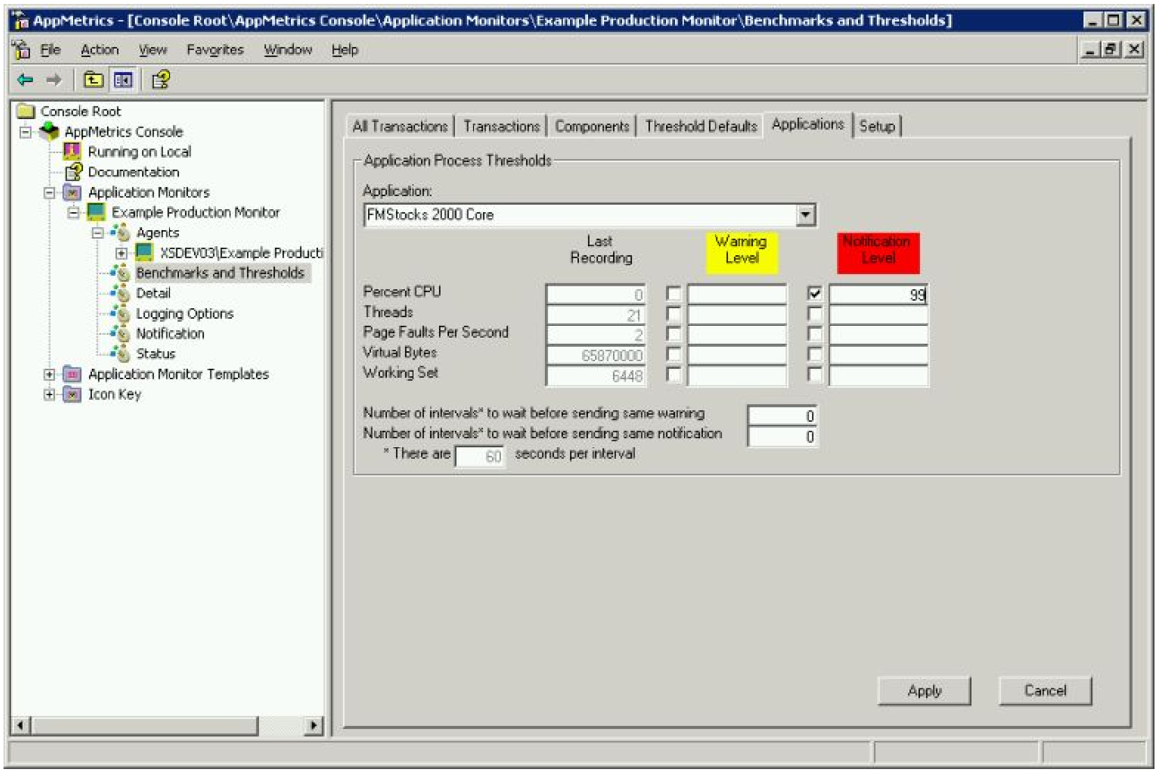
FMStocks_Bus.Account

	Last Recording		Current Benchmark	Warning Level	Notification Level
Number Aborted	0	-->	0	50 %	75 %
Number Started	10	-->	10	50 %	75 %
Duration (ms)	58	-->	58	50 %	75 %

5. The **Notification Level** fields are set to 75% by default. This means that if the value of the metric increases by more than 75%, a notification will be delivered. You may choose to change this value to a different setting if you wish.

6. Repeat for each component you wish to enable hung component handling.

7. Select the **Applications** page.



Application Process Thresholds

Application: FMStocks 2000 Core

	Last Recording		Warning Level		Notification Level
Percent CPU	0	☐		☑	99
Threads	21	☐		☐	
Page Faults Per Second	2	☐		☐	
Virtual Bytes	65870000	☐		☐	
Working Set	6448	☐		☐	

Number of intervals* to wait before sending same warning: 0

Number of intervals* to wait before sending same notification: 0

* There are 60 seconds per interval

8. The ***Applications*** page allows application thresholds to be set. In the above example, a Percent CPU value of 99 was entered for the **Notification Level**, which would cause a notification to be delivered if the application’s CPU usage reached or exceeded that value.

9. Once the thresholds have been set, select the **Apply** button.

Chapter 4. Configuring Notification Delivery

There are four different delivery mechanisms used to deliver notifications.

- SMTP mail
- Windows Event Log
- Component
- SNMP Manager

This section will briefly describe how to setup the notification delivery mechanisms in order to detect hung components and applications. For more detailed information, please refer to the AppMetrics documentation, Chapter 3, page 3-33.

Configuring SMTP Mail Notifications

The following steps will configure SMTP Mail delivery of AppMetrics notifications. Before proceeding, check to see if either of the two conditions below is true;

- The machine running AppMetrics Manager has no access to an e-mail server.
- The machine running AppMetrics Manager has access to an e-mail server, but the e-mail server does not allow relaying.

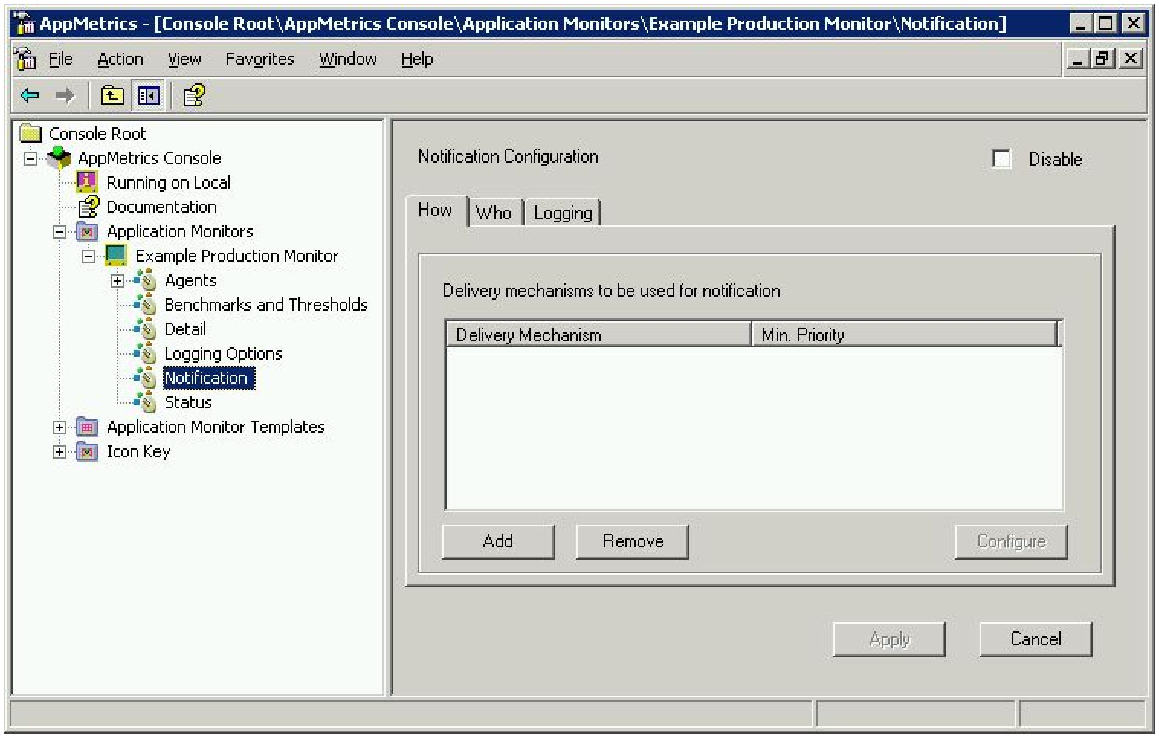
If any of the above conditions are true, please refer to the AppMetrics documentation, Chapter 3, page 3-33 before proceeding.

The following information will need to be determined prior to configuring SMTP Notifications;

- Email account to be used as the sender (if the server has an e-mail account, the use of that account may be more convenient in situations with multiple servers)
- Name of the SMTP host mail server
- SMTP port (usually port 25) on the AppMetrics Manager machine.

Check with your SMTP server administrator to identify these settings.

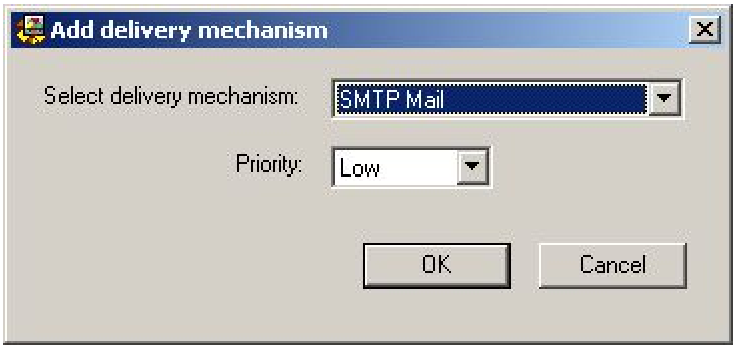
1. Click on the **Notification** node of the production monitor.


AppMetrics - [Console Root\AppMetrics Console\Application Monitors\Example Production Monitor\Notification]
File
Action
View
Favorites
Window
Help
Console Root
AppMetrics Console
Running on Local
Documentation
Application Monitors
Example Production Monitor
Agents
Benchmarks and Thresholds
Detail
Logging Options
Notification
Status
Application Monitor Templates
Icon Key
Notification Configuration
How
Who
Logging
Delivery mechanisms to be used for notification
Add
Remove
Configure
Apply
Cancel

☐ Disable

Delivery Mechanism	Min. Priority

2. Click on the **Add** button on the ***Notification Configuration How*** page.

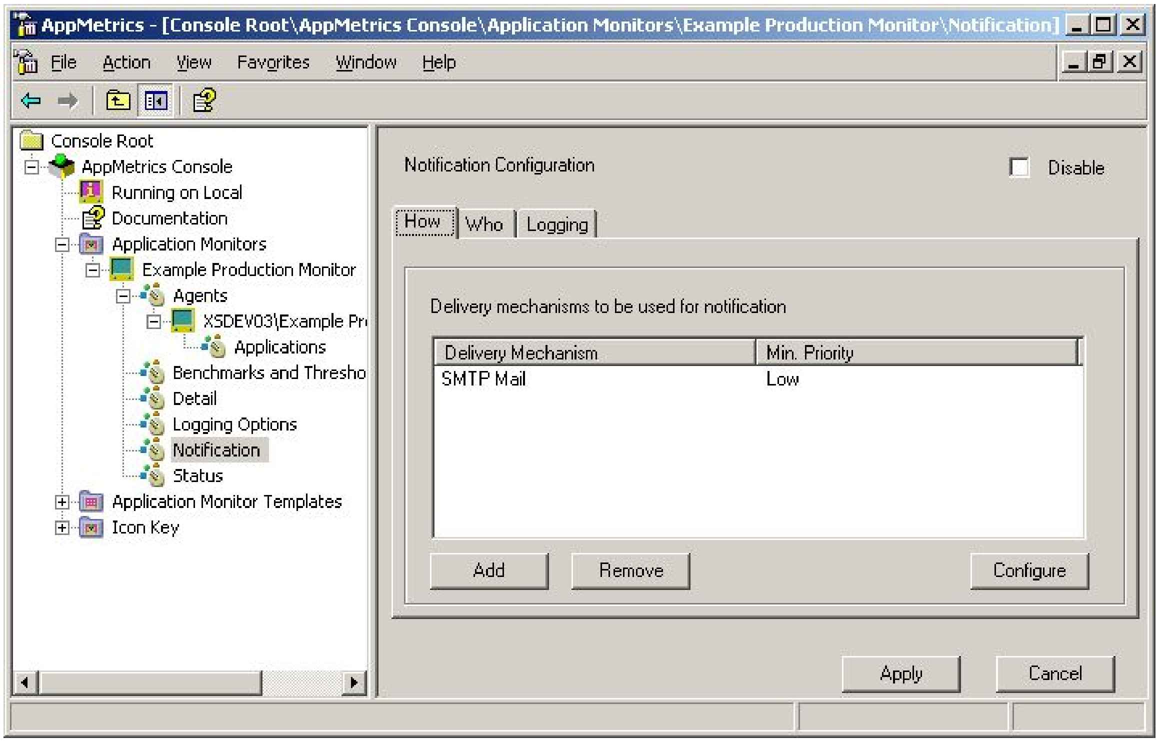


Select delivery mechanism: SMTP Mail

Priority: Low

3. Select **SMTP Mail** from the **Select Delivery Mechanism** combo box.

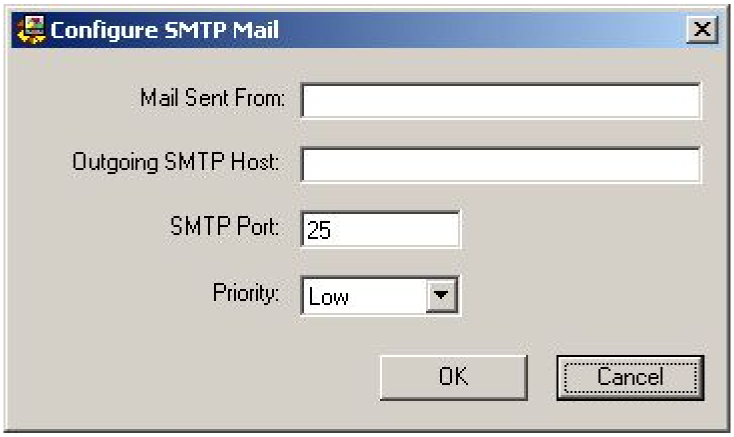
4. Click **OK**.



Delivery mechanisms to be used for notification

Delivery Mechanism	Min. Priority
SMTP Mail	Low

5. Select the **SMTP Mail** item in the **Delivery Mechanism** list box, and then click the **Configure** button.



Mail Sent From:

Outgoing SMTP Host:

SMTP Port: 25

Priority: Low

6. In the ***Configure SMTP Mail*** dialog box, enter the email account to be used as the sender in the **Mail Sent From** field. The account should be the full email address of the sender, i.e. sender@yourorg.com.

7. Enter the name of the SMTP mail server in the **Outgoing SMTP Host** field.

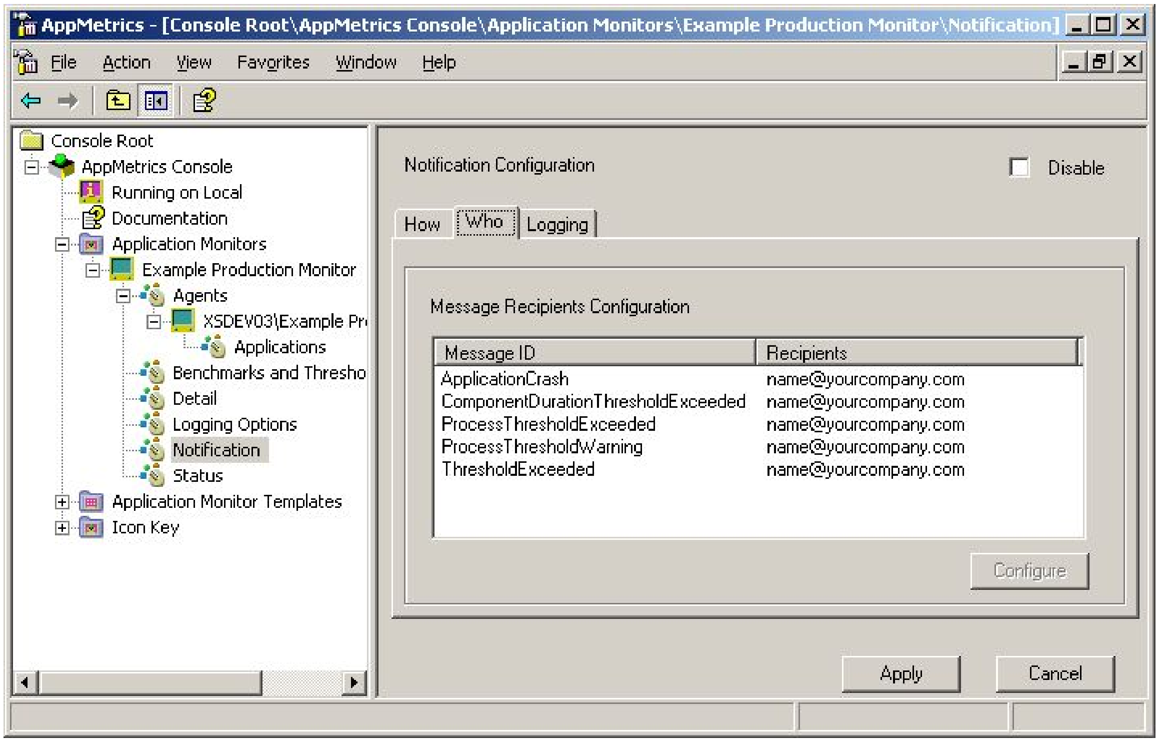
Note: If a SMTP virtual was configured as described in Chapter 3 of the AppMetrics documentation, enter the name of the local machine.

Otherwise, enter the name of your organization's email server.

8. Enter the SMTP port number in the **SMTP Port** field if it differs from the default value of port 25.

9. Click **OK**.

10. Select the ***Notification Configuration Who*** property page.



AppMetrics - [Console Root\AppMetrics Console\Application Monitors\Example Production Monitor\Notification]

- Console Root
  - AppMetrics Console
    - Running on Local
    - Documentation
    - Application Monitors
      - Example Production Monitor
        - Agents
          - XSDEV03\Example Pr
            - Applications
        - Benchmarks and Thresho
        - Detail
        - Logging Options
        - Notification
        - Status
    - Application Monitor Templates
    - Icon Key

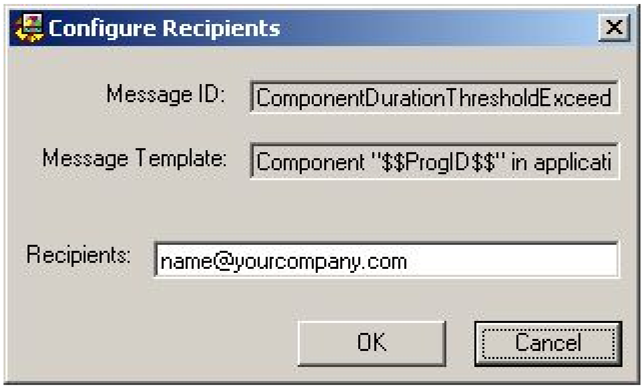
Notification Configuration

□ Disable

Message Recipients Configuration

Message ID	Recipients
ApplicationCrash	name@yourcompany.com
ComponentDurationThresholdExceeded	name@yourcompany.com
ProcessThresholdExceeded	name@yourcompany.com
ProcessThresholdWarning	name@yourcompany.com
ThresholdExceeded	name@yourcompany.com

11. From the ***Message Recipients Configuration*** list box, select the **ComponentDurationThresholdExceeded** item, and then click the **Configure** button.



Configure Recipients

Message ID: ComponentDurationThresholdExceed

Message Template: Component "$$ProgID$$" in applicati

Recipients: name@yourcompany.com

12. In the ***Configure Recipients*** dialog box, enter the email address of the person to be notified when a hung component is detected into the **Recipients** field.

13. Click **OK**.

14. Select the ProcessThresholdExceeded item from the ***Message Recipients Configuration*** list box, and then click the Configure button.

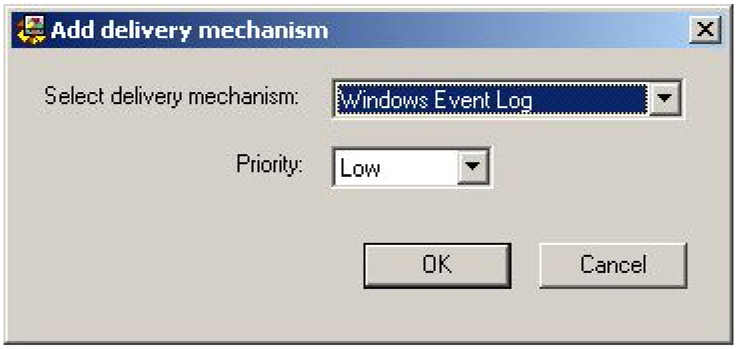
15. In the ***Configure Recipients*** dialog box, enter the email address of the person to be notified when a hung component is detected into the **Recipients** field.

16. Click **OK**.

Configuring Windows Event Log Notifications

Enabling Windows Event Log Notifications will cause AppMetrics notifications to be delivered to the Windows Application Event Log.

1. From the ***Notification Configuration How*** property page, select **Add**.



Select delivery mechanism: Windows Event Log

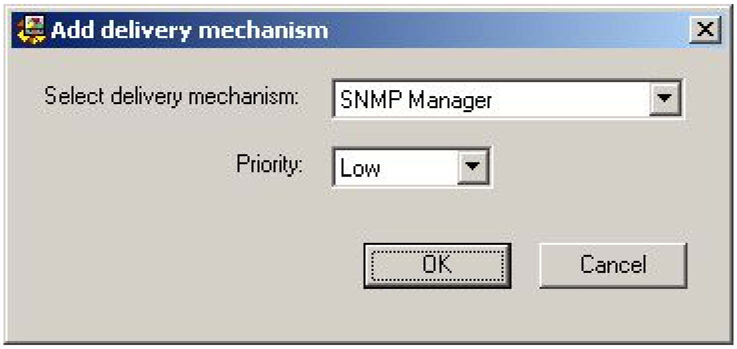
Priority: Low

2. Select the **Windows Event Log** from the **Select delivery mechanism** combo box, and then click **OK**.

Configuring SNMP Notifications

SNMP management must be configured prior to SNMP notification delivery. Please refer to Chapter 3 of the AppMetrics documentation, page 3-33, for more information.

1. From the ***Notification Configuration How*** property page, select **Add.**

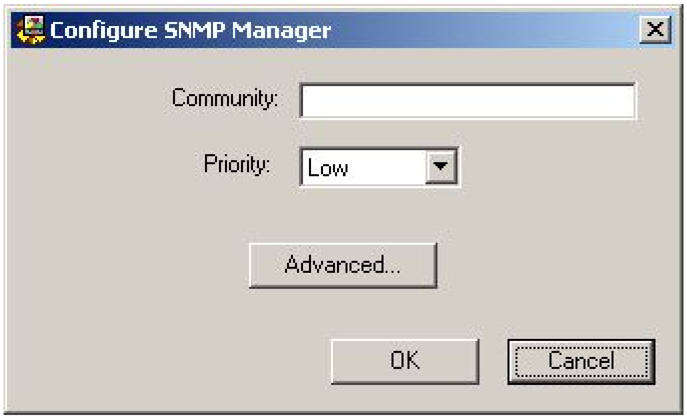


Select delivery mechanism: SNMP Manager

Priority: Low

2. Select the **SNMP Manager** from the **Select delivery mechanism** combo box, and then click **OK**.

3. Select the **SNMP Manager** item in the **Delivery Mechanism** list box, and then click the **Configure** button.



Priority: Low

4. Enter the SNMP community name into the **Community** field in the **Configure SNMP Manager** dialog box.

5. Select **OK**.

Chapter 5. Installing the Example Component Script

The following section describes how to install and configure the XSNotification.wsc script. This script will log alerts generated by AppMetrics and is provided as an example of how to create a custom script that will respond to AppMetrics notifications. It is installed in the \Program Files\Xtremesoft\AppMetrics for Transactions folder at install time.

Installing XSNotification.wsc

1. Ensure that the XSNotification.wsc file is present in the \Program Files\Xtremesoft\AppMetrics for Transactions folder.

2. You can choose to use the default ProgID for the script, **XSNotification.CustomAlert,** or use one of your own choosing. You will need to edit the file and change the value in the **progid** field if you elect to use your own ProdID.

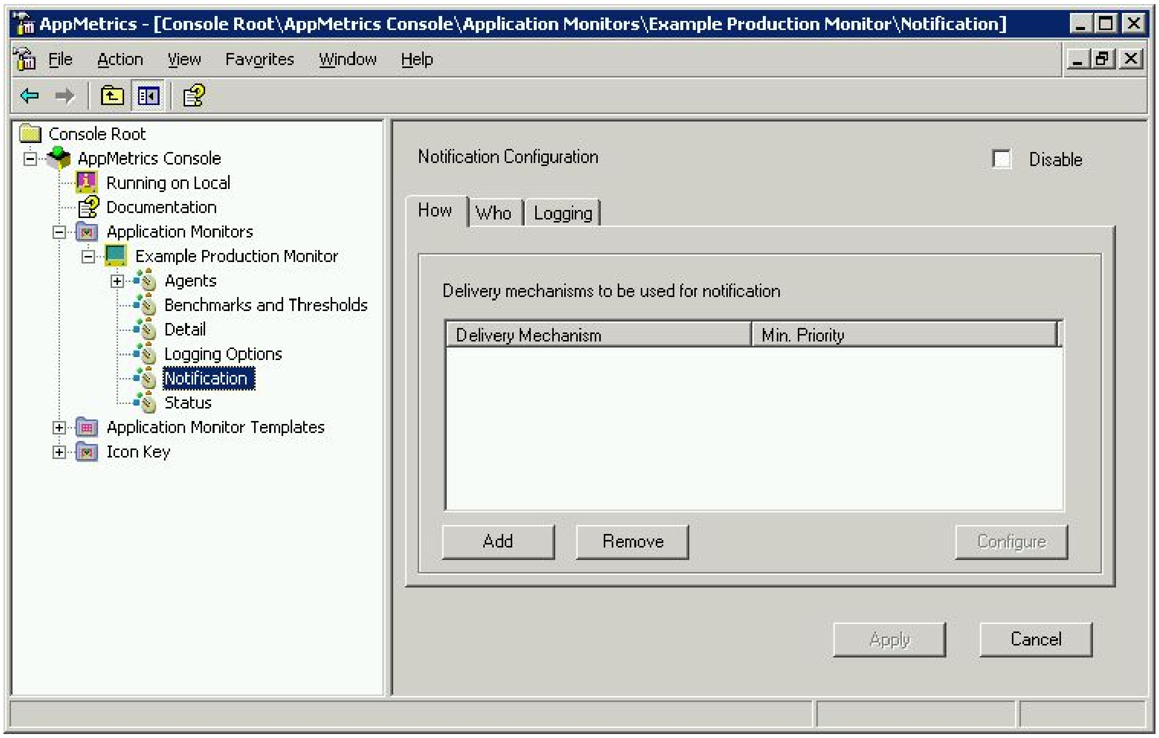
3. You can use the default ClassID for this component, or use a tool such as Microsoft's GUIDGEN in order create a unique GUID and use that generated GUID for the ClassID. If you choose to generate the GUID, you will need to edit the file and change the value in the **classid** field.

4. The value used for the ProgID in the file needs to be recorded in order to register that value in the **Configure Component** dialog in AppMetrics.

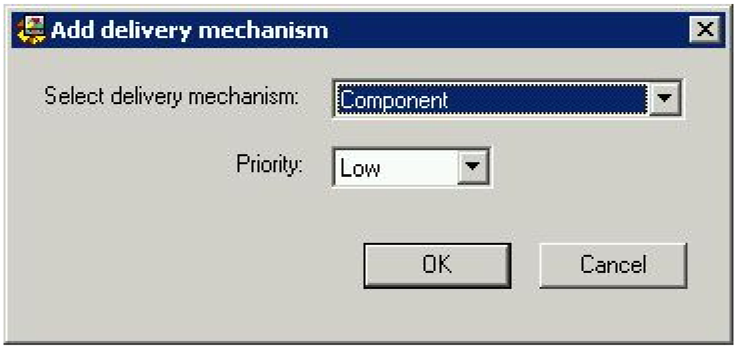
5. Right click the file, and select **Register**.

Configuring AppMetrics for XSNotification.wsc

1. Click on the **Notification** node of the production monitor.

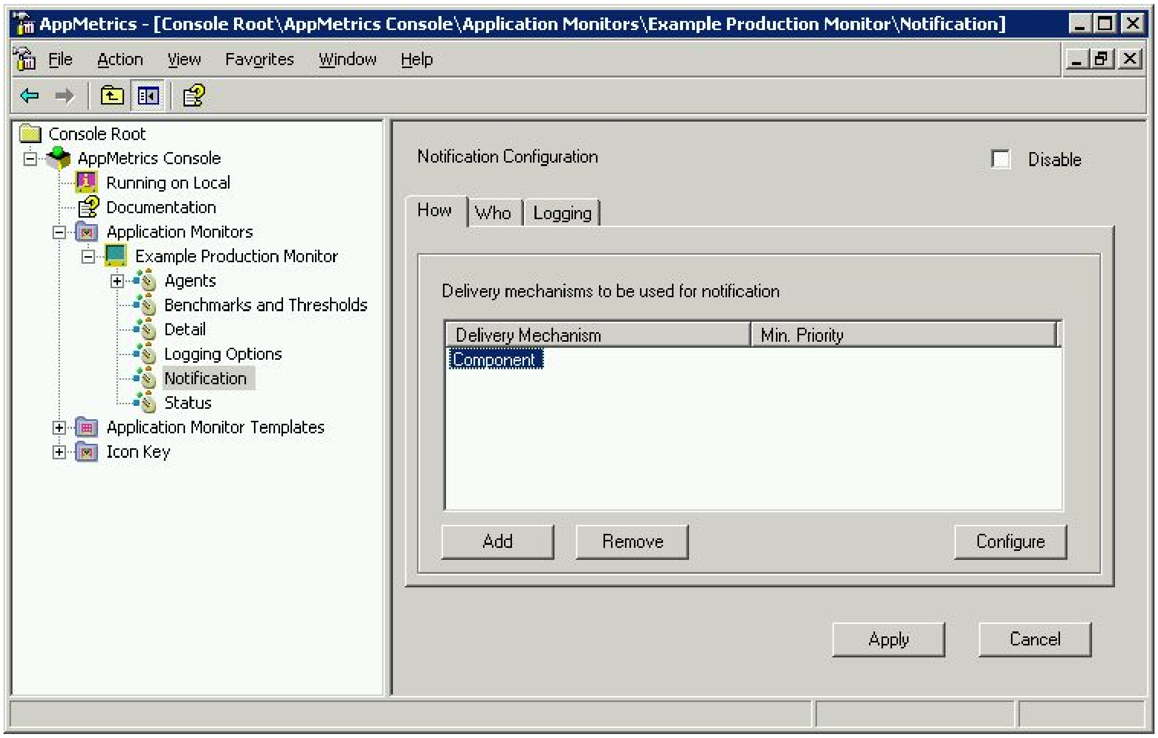

AppMetrics - [Console Root\AppMetrics Console\Application Monitors\Example Production Monitor\Notification]
File Action View Favorites Window Help
Console Root
AppMetrics Console
Running on Local
Documentation
Application Monitors
Example Production Monitor
Agents
Benchmarks and Thresholds
Detail
Logging Options
Notification
Status
Application Monitor Templates
Icon Key
Notification Configuration
Disable
How
Who
Logging
Delivery mechanisms to be used for notification
Delivery Mechanism
Min. Priority
Add
Remove
Configure
Apply
Cancel

2. Click on the **Add** button on the ***Notification Configuration How*** page.



3. Select **Components** from the **Select Delivery Mechanism** combo box.

4. Click on the **OK** button on the ***Add delivery mechanism*** dialog box.

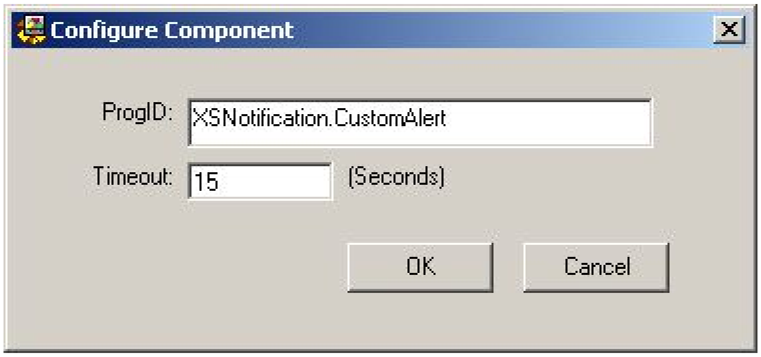


Delivery mechanisms to be used for notification

Delivery Mechanism	Min. Priority
Component	

5. Select the **Configure** button on the ***Notification Configuration How*** page.

6. Enter the appropriate **ProdID** in the ***Configure Component*** dialog box. The default **ProdID** for the XSNotificication.wsc script is XSNotification.CustomAlert.

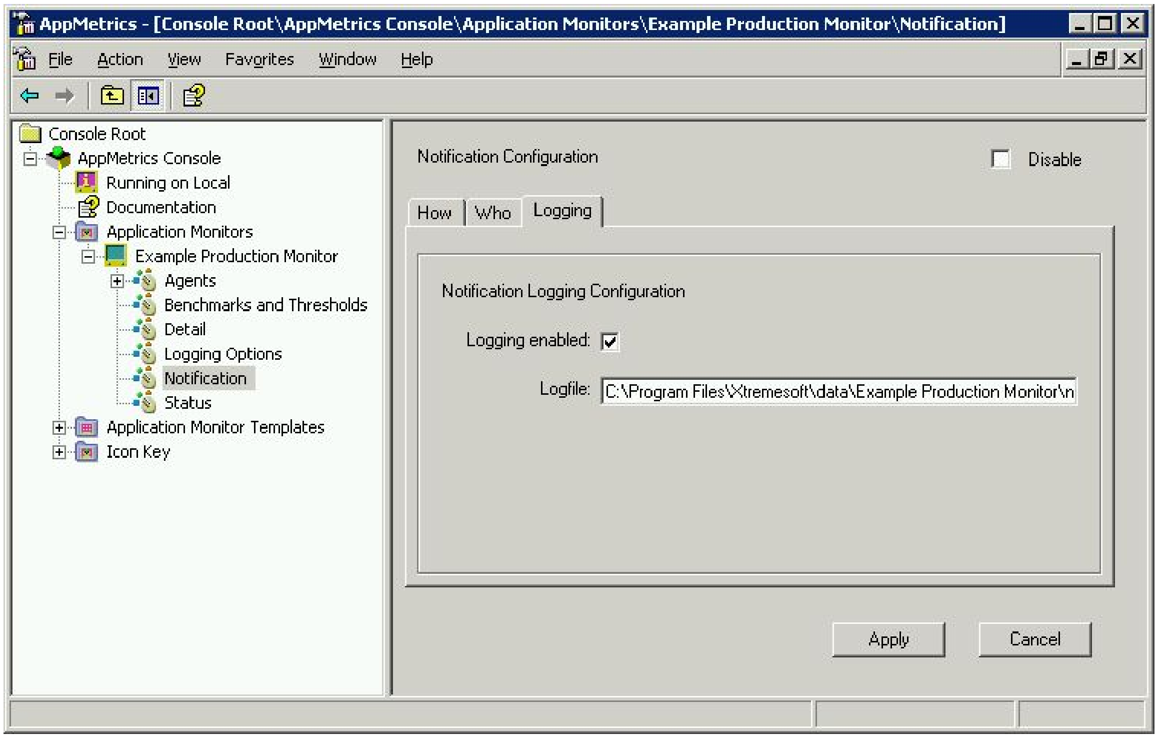


ProgID: XSNotification.CustomAlert

Timeout: 15 (Seconds)

7. Select **OK** on the ***Configure Component*** dialog box.

8. If it is desired to log the alerts generated, you may choose to enable logging by selecting the ***Notification Configuration Logging*** page and checking the **Logging enabled** checkbox.



Notification Configuration

□ Disable

Notification Logging Configuration

Logging enabled: ☑

Logfile: C:\Program Files\Xtremesoft\data\Example Production Monitor\n

9. Select the **Apply** button on the ***Notification Configuration How*** page.

Chapter 6. Enabling hung component handling

After ensuring that the thresholds are set properly, start the Application Monitor. As thresholds are met, the script will shut down the COM+ Application on the application server where the event occurred. COM+ will restart the COM+ application when a call is made to that application.

See the AppMetrics documentation, Chapter 3, pages 33-39 for more information.

In order to configure AppMetrics so that hung components are automatically restarted, please refer to the following section, ***Using AppMetrics to Restart Hung Components and Applications***.

Using AppMetrics to Restart Hung Components and Applications

In order to use AppMetrics to detect hung components, a Production Monitor must be created and configured in order to monitor the desired group of applications. Additionally, a special windows script, XSHungComponent.wsc, needs to installed and configured. Once the script is installed, the application thresholds need to be configured in order for the alerts to be generated when a component is hung.

Chapter 7. Installing the Hung Component Script

The following section describes how to install and configure the XSHungComponent.wsc script, which will handle alerts generated by AppMetrics and shutdown the appropriate COM+ application in order for that application to be restarted.

Installing XSHungComponent.wsc

1. Copy the XSHungComponent.wsc file to the C:\Program Files\Xtremesoft\AppMetrics for Transactions folder.

   Note: XSHungComponent.wsc is delivered in compressed format. You will need to first extract the file using a tool such as WinZip.

2. You can choose to use the default ProgID for the hung component handler script, **XSNotification.HungComponentHandler,** or use one of your own choosing. You will need to edit the file and change the value in the **progid** field if you elect to use your own ProdID.

3. You can use the default ClassID for this component, or use a tool such as Microsoft's GUIDGEN in order create a unique GUID and use that generated GUID for the ClassID. If you choose to generate the GUID, you will need to edit the file and change the value in the **classid** field.

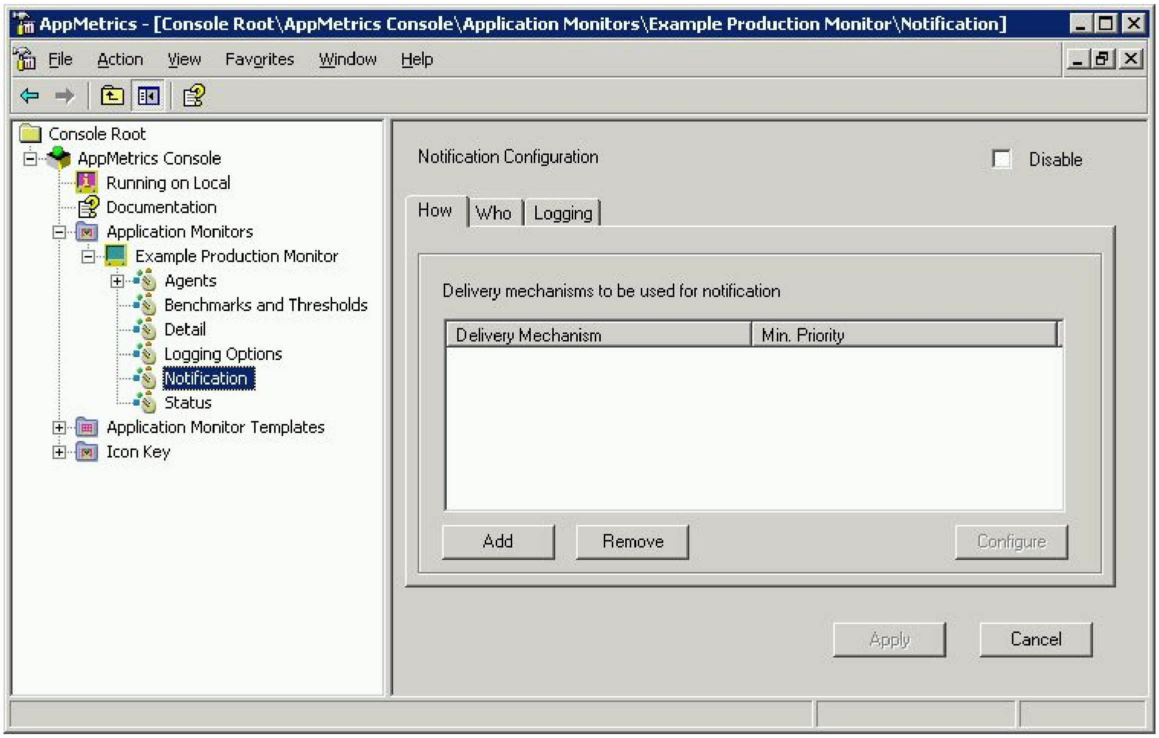
4. The value used for the ProgID in the file needs to be recorded in order to register that value in the **Configure Component** dialog in AppMetrics.

5. Right click the file, and select **Register**.

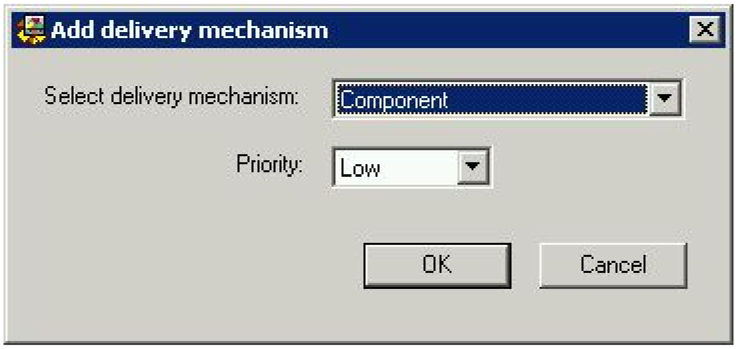
Configuring AppMetrics for XSHungComponent.wsc

1. Follow the steps outlined in the ***Setting up AppMetrics to Detect Hung Components*** section in order to create and configure a production monitor, to set the benchmarks and thresholds, and to configure notification delivery You may choose any notification delivery mechanism except for Component notification, as that will be used to trigger the XSHungComponent handler script.

2. Click on the **Notification** node of the production monitor.


AppMetrics - [Console Root\AppMetrics Console\Application Monitors\Example Production Monitor\Notification]
File
Action
View
Favorites
Window
Help
Console Root
AppMetrics Console
Running on Local
Documentation
Application Monitors
Example Production Monitor
Agents
Benchmarks and Thresholds
Detail
Logging Options
Notification
Status
Application Monitor Templates
Icon Key
Notification Configuration
Disable
How
Who
Logging
Delivery mechanisms to be used for notification
Delivery Mechanism
Min. Priority
Add
Remove
Configure
Apply
Cancel

3. Click on the **Add** button on the ***Notification Configuration How*** page.

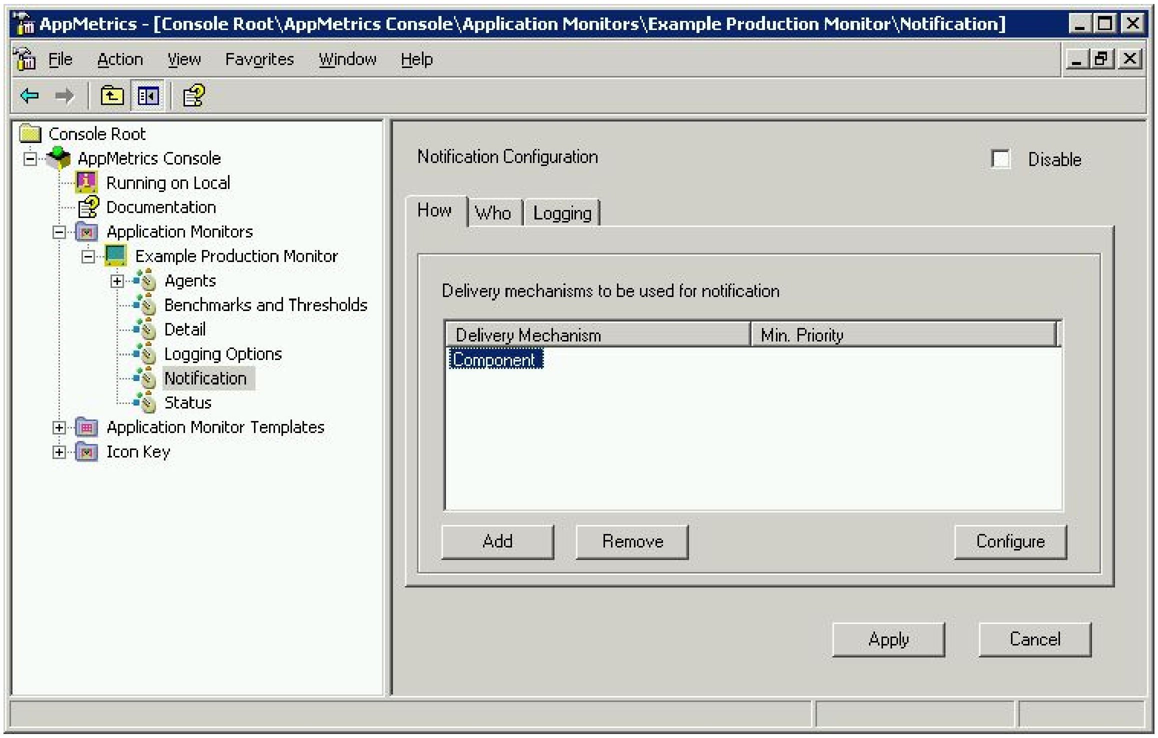


Select delivery mechanism: Component

Priority: Low

4. Select **Components** from the **Select Delivery Mechanism** combo box.

5. Click on the **OK** button on the ***Add delivery mechanism*** dialog box.



Notification Configuration

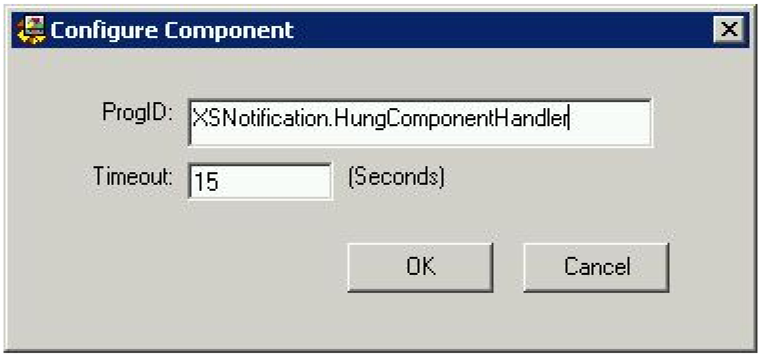
☐ Disable

Delivery mechanisms to be used for notification

Delivery Mechanism	Min. Priority
Component	

6. Select the **Configure** button on the ***Notification Configuration How*** page.

7. Enter the appropriate **ProdID** in the ***Configure Component*** dialog box. The default **ProdID** for the XSHungComponent.wsc script is XSNotification.HungComponentHandler.

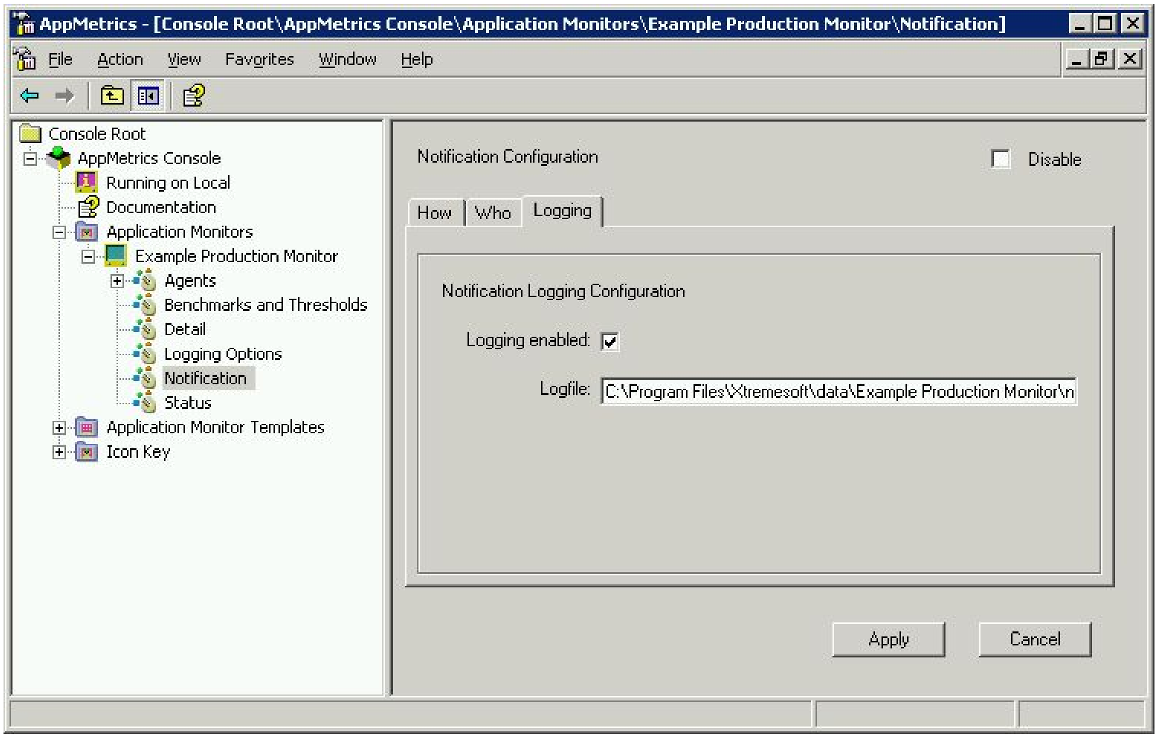


ProgID: XSNotification.HungComponentHandler

Timeout: 15 (Seconds)

8. Select **OK** on the ***Configure Component*** dialog box.

9. If it is desired to log the alerts generated, you may choose to enable logging by selecting the ***Notification Configuration Logging*** page and checking the **Logging enabled** checkbox.



AppMetrics - [Console Root\AppMetrics Console\Application Monitors\Example Production Monitor\Notification]
File
Action
View
Favorites
Window
Help
Console Root
AppMetrics Console
Running on Local
Documentation
Application Monitors
Example Production Monitor
Agents
Benchmarks and Thresholds
Detail
Logging Options
Notification
Status
Application Monitor Templates
Icon Key
Notification Configuration
Disable
How
Who
Logging
Notification Logging Configuration
Apply
Cancel

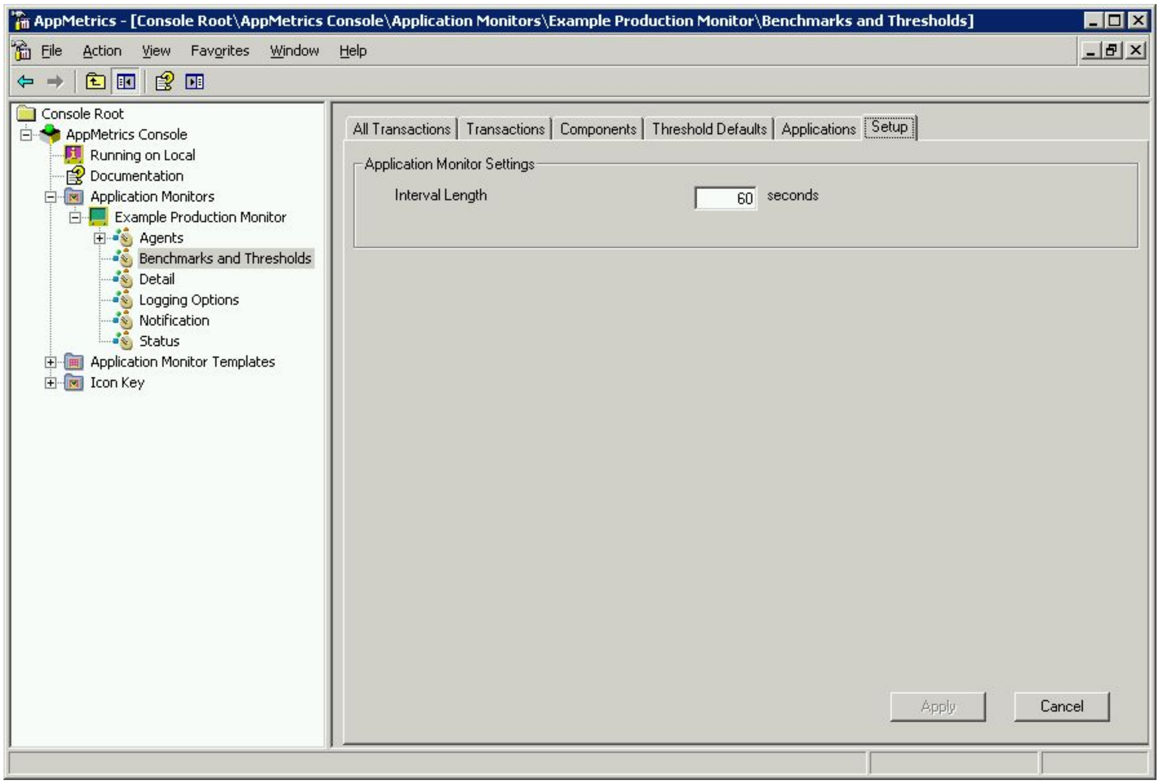
Notification Configuration □ Disable

Logging enabled: ☑

Logfile: C:\Program Files\Xtremesoft\data\Example Production Monitor\n

10. Select the **Apply** button on the ***Notification Configuration How*** page.

Once the monitor is started, AppMetrics will monitor the application(s) and report any thresholds that have been exceeded if notifications have been set accordingly. Any hung component will cause the HungComponent handler to be triggered if the duration of the hang exceeds one interval. Interval durations can be set by adjusting the value on the **Setup** screen located in **the Benchmarks and Thresholds** property pages, as shown below.



Application Monitor Settings

Interval Length 60 seconds

Note: Even though the component may exceed the duration set in the Component Thresholds screen, it needs to be hung past the end of an interval to be reported as a hung component. If the component duration exceeds the threshold set in the Component Thresholds screen but does not hang past the end of an interval, it will simple be reported as a duration threshold exceeded alert rather than a hung component alert, thus the hung component script will not be executed, since the component is not actually hung.

The default setting is 60 seconds, and is typically an optimal value. However, more frequent checks may be desired in certain applications, thus AppMetrics provides the means to modify this setting.