


Minimum Monitoring and Alerting Solution

Article



[Murray Oldfield](#) · Mar 15, 2018

 14m read

Minimum Monitoring and Alerting Solution

InterSystems Data Platform includes utilities and tools for system monitoring and alerting, however System Administrators new to solutions built on the InterSystems Data Platform (a.k.a Caché) need to know where to start and what to configure.

This guide shows the path to a minimum monitoring and alerting solution using references from online documentation and developer community posts to show you how to enable and configure the following;

1. **Caché Monitor:** Scans the console log and sends emails alerts.
2. **System Monitor:** Monitors system status and resources, generating notifications (alerts and warnings) based on fixed parameters and also tracks overall system health.
3. **Health Monitor:** Samples key system and user-defined metrics and compares them to user-configurable parameters and established normal values, generating notifications when samples exceed applicable or learned thresholds.
4. **History Monitor:** Maintains a historical database of performance and system usage metrics.
5. **pButtons:** Operating system and Caché metrics collection scheduled daily.

Remember this guide is a minimum configuration, the included tools are flexible and extensible so more functionality is available when needed. This guide skips through the documentation to get you up and going. You will need to dive deeper into the documentation to get the most out of the monitoring tools, in the meantime, think of this as a set of cheat sheets to get up and running.

1. Caché Monitor

The console log (`install-directory/mgr/cconsole.log`) must be monitored, either through third party tools that scan the log file or as we do here using the included Caché Monitor utility to send alerts to an email address.

The console log is the central repository for other monitoring tools including Caché System Monitor and Caché Health Monitor to write their alerts and notifications.

At a minimum configure Caché Monitor to send alerts to an email.

Caché Monitor is managed with the ^MONMGR utility.

Caché Monitor Basic set up

Caché Monitor scans the console log and generates notifications based on configurable message severity level. Notifications are sent by email to a list of recipients you configure. The default scan period is every 10 seconds, but this can be changed.

Tip: Configure Caché Monitor alert severity level to 1 (warning, severe and fatal entries). If you find you are getting too many alerts you can drop back to alert level 2 (severe and fatal entries).

When there is a series of entries within 60 seconds from a given process a notification is generated for the first entry only then suspended for one hour. So you must investigate problems when they arise. Because there are no new messages does not mean an event has passed. The exception to this rule is console log entries listed in [Caché Monitor Errors and Traps](#) which generate notifications for all entries.

[See online documentation for ^MONMGR for full configuration details.](#)

Caché Monitor Cheat sheet

There is not much to do to enable Caché Monitor. Ensure the Monitor is started, then set the email options.

```
%SYS>d ^MONMGR
```

```
1) Start/Stop/Update MONITOR
2) Manage MONITOR Options
3) Exit
Option? **1**
```

```
1) Update MONITOR
2) Halt MONITOR
3) Start MONITOR
4) Reset Alerts
5) Exit
Option? **3**
```

```
Starting MONITOR... MONITOR started
```

```
1) Update MONITOR
2) Halt MONITOR
3) Start MONITOR
4) Reset Alerts
5) Exit
Option? **<return>**
```

Set the alert severity level.

```
1) Start/Stop/Update MONITOR
2) Manage MONITOR Options
3) Exit
Option? **2**
```

Minimum Monitoring and Alerting Solution

Published on InterSystems Developer Community (<https://community.intersystems.com>)

- 1) Set Monitor Interval
- 2) Set Alert Level
- 3) Manage Email Options
- 4) Exit

Option? **2**

Alert on Severity (1=warning,2=severe,3=fatal)? 2 => **1**

Set email options, you may have to talk to your IT department to get the address of your email server. Any valid email address should work.

- 1) Set Monitor Interval
- 2) Set Alert Level
- 3) Manage Email Options
- 4) Exit

Option? **3**

- 1) Enable/Disable Email
- 2) Set Sender
- 3) Set Server
- 4) Manage Recipients
- 5) Set Authentication
- 6) Test Email
- 7) Exit

Option?

Make sure you test the email after set up (option 6).

Caché Monitor Example

The Caché System Monitor generates a severity 2 entry for high CPU utilisation which is sent to cconsole.log:

```
03/07/18-11:44:50:578 (4410) 2 [SYSTEM MONITOR] CPUusage Alert: CPUusage = 92, 95, 98
(Max value is 85).
```

An email is also sent to Caché Monitor email recipients with the same message as the console.log and the subject line:

```
[CACHE SEVERE ERROR yourserver.name.com:instancename] [SYSTEM MONITOR] CPUusage Alert
: CPUusage = 92, 95, 98 (Max value is 85).
```

Caché Monitor More Tips

There is also [another article on the developer community](#) with a comment by Aric West with the tip to embed more information in the sender email, for example rather than just a valid email setting sender or recipient to: "Some Name" \valid@emailAddress.com>

Caché System Monitor Tools

Minimum Monitoring and Alerting Solution

Published on InterSystems Developer Community (<https://community.intersystems.com>)

Caché System Monitor is the umbrella for a collection of monitoring tools and is configured through the ^%SYSMONMGR utility. As mentioned in the introduction for a minimum monitoring solution we will configure;

- System Monitor
- Health Monitor
- History Monitor

As a sidebar, yes, the *System Monitor* name is annoyingly overloaded with *Caché System Monitor* it's parent name, .

Caché System Monitor and Health Monitor notifications and alerts are sent to the console log allowing Caché Monitor (set up in the previous section) to generate email messages when they occur.

[All the deep detail on Caché System Monitor is in the online documentation.](#)

Warning Note: You also will see **Application Monitor** in the ^%SYSMONMGR menu. Application Monitor will not be configured in this guide. The tools and utilities shown in this guide have negligible impact on system performance, however Application Monitor does have some classes that are an exception to this rule. [For more details see documentation for ^PERFMON utility.](#) If you use Application Monitor you must test on non-production systems first as there can be a significant performance impact running ^PERFMON for any length of time.

2. System Monitor

From the documentation: "System Monitor samples important system status and resource usage indicators, such as the status of ECP connections and the percentage of the lock table in use, and generates notifications (alerts, warnings, and "status OK" messages) based on fixed statuses and thresholds."

There is a list of [System Monitor Status and Resource Metrics in the documentation](#). For example: Journal Space (Available space in the journal directory):

- less than 250 MB = *warning*
- less than 50 MB = *alert*
- greater than 250 MB after warning/alert = *OK*

System Monitor Basic Set Up

System Monitor alerts and warnings are written to the console log, so ensure that Caché Monitor is set up to send email alerts (previous section).

System Monitor is managed using the ^%SYSMONMGR utility. By default, the System Monitor is always running when the instance is running; it can be stopped using ^%SYSMONMGR but will start automatically again when the instance next starts.

By default the System Monitor has the following settings, which can be changed:

- Gets sensor metrics every 30 seconds.
 - Writes only alerts, warnings and messages to the System Monitor log.

System Monitor also maintains a single overall system health state which can be queried or is available when you run commands such as ccontrol list:

- Green (OK)
- Yellow (warning)

- Red (alert)

System Monitor Cheat Sheet

Nothing really to do for a minimal monitoring solution as it is always running when the instance is running.

3. Health Monitor

From the documentation: “Caché Health Monitor monitors a running Caché instance by sampling the values of a broad set of key metrics during specific periods and comparing them to configured parameters for the metric and established normal values for those periods; if sampled values are too high, Health Monitor generates an alert or warning. For example, if CPU usage values sampled by Health Monitor at 10:15 AM on a Monday are too high based on the configured maximum value for CPU usage or normal CPU usage samples taken during the Monday 9:00 AM to 11:30 AM period, Health Monitor generates a notification.”

[Health Monitor samples 41 system sensors, the list and defaults are in the documentation.](#)

Health Monitor alerts (severity 2) and warnings (severity 1) are written to the console log. Health Monitor generates:

- An alert if three consecutive readings of a sensor during a period are greater than the sensor maximum value.
- A warning if five consecutive readings of a sensor during a period are greater than the sensor warning value.

An alert will be generated immediately for a sensor that has an entry set for maximum or warning value, even when Health Monitor itself is not enabled. For example CPU has a configured maximum of 85 and warning value of 75, so when there have been 5 consecutive CPU utilisation measurements over 75% the following notification is sent to the console log;

```
1 [SYSTEM MONITOR] CPUusage Warning: CPUusage = 83 ( Warnvalue is 75).
```

Other sensors require metrics to be collected for long enough to create a *chart*. A chart is needed to evaluate the mean value for a metric and therefore the standard deviation (σ) so that alerts can be sent when values fall out of normal range. Metrics are collected within a *period*. There are 63 standard periods, an example of a period is Monday 9:00 AM to 11:30 AM. Periods may be changed.

Health Monitor Basic Set Up

Health Monitor does not start automatically, to enable this use settings in `^%SYSMONMGR`.

By default Health Monitor waits 10 minutes after Caché startup to allow the system to reach normal operation, this can be changed if needed.

Caché Health Monitor Sensor Objects ship with default values. For example, as above, defaults for *CPUPct* (System CPU usage %) is: base 50, Maximum 90, Warning 80.

You might be more conservative and want to change these values, using `^%SYSMONMGR` you can change the values, for example; Maximum 85, Warning 70. Now when the CPU is being thrashed at 99% we see;

```
2 [SYSTEM MONITOR] CPUusage Alert: CPUusage = 99, 99, 99 (Max value is 85).
```

Health Monitor cheat sheet

The cheat sheet is quite long, it appears after the Summary.

4. History Monitor

[David Loveluck has a great post on community.](#) Follow the instructions in that post to start History Monitor and start collecting and reviewing metrics.

5. pButtons

The pButtons utility generates a readable HTML performance report with operating system and Caché metrics from log files it creates. Performance metrics output by pButtons can be extracted, charted and reviewed. For example, a chart of CPU utilisation or Caché database access across the day.

Running a daily 24 hour pButtons collection is a simple but vital way to collect metrics for troubleshooting. pButtons is also very useful for trend analysis. The following Community articles have details of pButtons and instructions for scheduling it to run daily:

[InterSystems Data Platforms and performance – Part 1](#)

As noted in the article a 30 second collection interval is fine for trend analysis and 24 hour reporting.

There are also instructions for ensuring you have the most up to date version of pButtons, even if you are not running the latest version of Caché:

[InterSystems Data Platforms and performance – how to update pButtons.](#)

Although pButtons is primarily a support tool, you can gain valuable insights of your systems usage by quickly charting and graphing collected metrics:

[Yape - Yet another pButtons extractor \(and automatically create charts\)](#)

Summary

This post has just scratched the surface of the options for monitoring, for example the Health Monitor will work with defaults, but over time you will want to explore the options to customise to your application profile.

Where to next?

As we saw System Monitor and Health Monitor utilities we configured send alerts to cconsole.log as a central reporting location. We used Caché Monitor to surface those alerts to email. There are third party tools that scrape logs and consume unstructured log data that you may be using in your organisation already, and there is no reason you could not use them instead.

Many customers I see today are virtualised on VMware. If you are using vSphere consider using Log Insight for monitoring the console log. At the date of writing this post (March 2018) for each instance of vCenter Server 6.0 you own you are entitled to a free 25 OSI license of vRealize Log Insight for vCenter. Log insight is a tool for reading unstructured data and is used for log management & analytics — for example you can use it with cconsole.log — If this interests you contact VMware for more information. In the meantime I am planning a future post to show Log Insight working with cconsole.log.

If your collecting metrics you still have to look at them and know what they mean, I will keep writing posts to show how to interpret the information presented, especially performance metrics.

Application Performance Monitoring

David Loveluck has a series of posts on the community on Application Performance Monitoring, search for APM on community, or [start here](#).

Appendix: Health Monitor Cheat Sheet

This cheat sheet shows the process to start the Health Monitor we looked at in section 3 and walks through editing a sensor threshold values.

First let us start the Health Monitor.

```
%SYS>**d ^%SYSMONMGR**
```

```
1) Start/Stop System Monitor
2) Set System Monitor Options
3) Configure System Monitor Classes
4) View System Monitor State
5) Manage Application Monitor
6) Manage Health Monitor
7) View System Data
8) Exit
Option? **6**
```

```
1) Enable/Disable Health Monitor
2) View Alerts Records
3) Configure Health Monitor Classes
4) Set Health Monitor Options
5) Exit
Option? **1**
```

```
Enable Health Monitor? No => **yes**
```

```
Health Monitor is Enabled. Stop and restart System Monitor to run Health Monitor
```

As we can see from the message navigate back to the first menu or start ^%SYSMONMGR again to to stop and start System Monitor to complete the process.

```
%SYS>**d ^%SYSMONMGR**
```

```
1) Start/Stop System Monitor
2) Set System Monitor Options
etc...
```

We will go ahead here with an example of editing the CPUPct threshold.

```
%SYS>**d ^%SYSMONMGR**
```

```
1) Start/Stop System Monitor
2) Set System Monitor Options
3) Configure System Monitor Classes
```

Minimum Monitoring and Alerting Solution

Published on InterSystems Developer Community (<https://community.intersystems.com>)

```
4) View System Monitor State
5) Manage Application Monitor
6) Manage Health Monitor
7) View System Data
8) Exit
Option? **6**
```

```
1) Enable/Disable Health Monitor
2) View Alerts Records
3) Configure Health Monitor Classes
4) Set Health Monitor Options
5) Exit
Option? **3**
```

```
1) Activate/Deactivate Rules
2) Configure Periods
3) Configure Charts
4) Edit Sensor Objects
5) Reset Defaults
6) Exit
Option? **4**
```

Lets have a look at all the sensors first;

```
1) List Sensor Objects
2) Edit Sensor Object
3) Exit
Option? **1**
```

Sensor	Base	Max	Max M	Warn	Warn M
--	----	---	-----	-----	-----
CPUPct	50	80	0	70	0
CPUusage	50	85	0	75	0
CSPActivity	100	0	2	0	1.6
:					
: <Big list of sensors goes here>					
:					
WDWIJTime	60	0	2	0	1.6
WDWriteSize	1024	0	2	0	1.6

```
1) List Sensor Objects
2) Edit Sensor Object
3) Exit
Option? **2**
```

Cannot configure while System Monitor is running.

```
1) List Sensor Objects
2) Edit Sensor Object
3) Exit
```

D'oh, we need to go back and disable Health Monitor and System Monitor first.

```
1) Enable/Disable Health Monitor
2) View Alerts Records
3) Configure Health Monitor Classes
```


Minimum Monitoring and Alerting Solution

Published on InterSystems Developer Community (<https://community.intersystems.com>)

```
4) Set Health Monitor Options
5) Exit
Option? **1**
```

```
Disable Health Monitor? No => **yes**
```

```
Health Monitor is Disabled. Stop and restart System Monitor to halt Health Monitor
```

```
1) Enable/Disable Health Monitor
2) View Alerts Records
3) Configure Health Monitor Classes
4) Set Health Monitor Options
5) Exit
Option?**<return>**
```

```
1) Start/Stop System Monitor
2) Set System Monitor Options
3) Configure System Monitor Classes
4) View System Monitor State
5) Manage Application Monitor
6) Manage Health Monitor
7) View System Data
8) Exit
Option? **1**
```

```
1) Start System Monitor
2) Stop System Monitor
3) Exit
Option? **2**
```

```
Stopping System Monitor... System Monitor stopped
```

OK, Health Monitor and System Monitor are stopped. Now navigate back to the Health Monitor and edit a sensor object.

```
1) Start System Monitor
2) Stop System Monitor
3) Exit
Option?**<return>**
```

```
1) Start/Stop System Monitor
2) Set System Monitor Options
3) Configure System Monitor Classes
4) View System Monitor State
5) Manage Application Monitor
6) Manage Health Monitor
7) View System Data
8) Exit
Option? **6**
```

```
1) Enable/Disable Health Monitor
2) View Alerts Records
3) Configure Health Monitor Classes
4) Set Health Monitor Options
5) Exit
Option? **3**
```

```
1) Activate/Deactivate Rules
```

```
2) Configure Periods
3) Configure Charts
4) Edit Sensor Objects
5) Reset Defaults
6) Exit
Option? **4**
```

```
1) List Sensor Objects
2) Edit Sensor Object
3) Exit
Option? **2**
```

Enter the sensor name if you know it, else "?" for a list.

Sensor? ?

```
Num  Sensor                      Threshold
  1)  CPUPct
  2)  CPUUsage
:
: <Big list of sensors goes here>
:
46)  WDWIJTime
47)  WDWwriteSize
```

```
Sensor? **1** CPUPct
Base? 50 => **<return>**
Enter either an Alert Value or a Multiplier
Alert Value? 80 => **85**
Setting Max Multiplier and Warn Multiplier to 0. Enter a Warn Value
Warn Value? 70 => **75**
Sensor object CPUPct updated.
Base          50
MaxMult       0
AlertValue    85
WarnMult      0
WarnValue     75
```

```
1) List Sensor Objects
2) Edit Sensor Object
3) Exit
```

Now back out and enable Health Monitor and start System Monitor.

```
Option? **<return>**
```

```
1) Activate/Deactivate Rules
2) Configure Periods
3) Configure Charts
4) Edit Sensor Objects
5) Reset Defaults
6) Exit
Option? **<return>**
```

```
1) Enable/Disable Health Monitor
2) View Alerts Records
```

Minimum Monitoring and Alerting Solution

Published on InterSystems Developer Community (<https://community.intersystems.com>)

```
3) Configure Health Monitor Classes
4) Set Health Monitor Options
5) Exit
Option? **1**
```

```
Enable Health Monitor? No => **yes**
```

```
Health Monitor is Enabled. Stop and restart System Monitor to run Health Monitor
```

```
1) Enable/Disable Health Monitor
2) View Alerts Records
3) Configure Health Monitor Classes
4) Set Health Monitor Options
5) Exit
Option?**<return>**
```

```
1) Start/Stop System Monitor
2) Set System Monitor Options
3) Configure System Monitor Classes
4) View System Monitor State
5) Manage Application Monitor
6) Manage Health Monitor
7) View System Data
8) Exit
Option? **1**
```

```
1) Start System Monitor
2) Stop System Monitor
3) Exit
Option? **1**
```

```
Starting System Monitor... System Monitor started
```

OK, you are good to go!

[#Beginner](#) [#Best Practices](#) [#InterSystems Business Solutions and Architectures](#) [#InterSystems Data Platform Blog](#)
[#System Administration](#) [#Caché](#)

110 4 5 1 1,139

Source URL: <https://community.intersystems.com/post/minimum-monitoring-and-alerting-solution>