**WebSphere Application Server Traditional Certificate Expiration Monitor Data Collection**

**Written by L3 support**

* Hiroko Takamiya (<mailto:htakamiy@us.ibm.com>)

**Version History**

* V1 (July 18th, 2019): First version. Providing with PH09722 test fix

**Contents**

[1 Collecting data for Certificate Expiration Monitor issue 2](#_Toc14439495)

[1.1 Server startup trace (Optional) 2](#_Toc14439496)

[1.2 Diagnostic Plan 2](#_Toc14439497)

[1.2.1 Sample scripts 2](#_Toc14439498)

[1.2.2 Enabling the diagnositic plan 3](#_Toc14439499)

[1.2.3 Canceling the diagnostic plan 3](#_Toc14439500)

[1.2.4 Querying the diagnostic plan 3](#_Toc14439501)

[2 Collecting data 4](#_Toc14439502)

[2.1.1 Reading the startup trace 4](#_Toc14439503)

[3 Newly added messages 5](#_Toc14439504)

# Collecting data for Certificate Expiration Monitor issue

The certificate expiration monitor runs once a month by default. The monitor is usually kicked off during peak hours. When an issue is suspected with the monitor, it is not easy for the system administrators and IBM support team to collect data, especially in the production environment.

At WebSphere 8.5.5.15[[1]](#footnote-1) and 9.0.0.8[[2]](#footnote-2). *Diagnostic Plan [[3]](#footnote-3) [[4]](#footnote-4)* feature was introduced. By using a properly configured diagnostic plan, WebSphere can automatically enable necessary trace options and disable the tracing by looking for messages as a trigger.

To take advantage of this feature, security APAR PH09722 adds more messages around the certificate expiration monitor function. These messages help trigger the diagnostic plan to start and stop collecting traces.

With both diagnostic plan and PH09722 in place, the system administrators can just set the diagnostic plan ahead of the time and wait for the certificate expiration monitor to run. Then whenever it is good time, upload the produced trace to IBM support.

The diagnostic plan can run on the deployment manager and the standalone application server. At this time, when running on the deployment manager, it does not produce trace for federated nodeagent and servers.

## Server startup trace (Optional)

The server startup trace can confirm the scheduler information is processed correctly at startup. If there is a chance to capture the startup trace, save a copy of security.xml before the startup, then configure following trace option prior to restart the server. Once the server is started fine and there’s no certificate related activity in SystemOut.log, turn off the trace and send the before/after security.xml and the startup trace to IBM Support.

com.ibm.ws.security.core.distSecurityComponentImpl=all:SSL=all

## Diagnostic Plan

This section shows how to enable, cancel and query the diagnostic plan for the certificate expiration monitor by the sample script provided via the support ticket.

### Sample scripts

It is handy to create diagnostic plans or download them. When the certificate expiration monitor is running on the dmgr, use the scripts ending with “Dmgr” below.

* setDiagPlanCertMonitorDmgr.py
* queryDiagPlanDmgr.py
* clearDiagPlanDmgr.py

When the certificate expiration monitor is running on the standalone application server, use the scripts ending with “AppSvr” below.

* setDiagPlanCertMonitorAppSvr.py
* queryDiagPlanAppSvr.py
* clearDiagPlanAppSvr.py

### Enabling the diagnositic plan

To enable the disgnostic plan for the certificate expiration monitor, run following wsadmin command with the jython script setDiagPlanCertMonitor.py.

# ./wsadmin.sh -username Administrator -password password -lang jython -f setDiagPlanCertMonitor.py

WASX7209I: Connected to process "dmgr" on node htakamiyDmgrCellManager01 using SOAP connector; The type of process is: DeploymentManager

Setting the Runtime Trace Output to Memory Buffer

Certificate Monitor diagplan is set

### Canceling the diagnostic plan

To cancel the disgnostic plan for the certificate expiration monitor, run following wsadmin command with the jython script clearDiagPlanDmgr.py.

# ./wsadmin.sh -username Administrator -password password -lang jython -f clearDiagPlanDmgr.py

WASX7209I: Connected to process "dmgr" on node htakamiyDmgrCellManager01 using SOAP connector; The type of process is: DeploymentManager

None

TRAS1102I: No diagnostic plan is set or running. To set a diagnostic plan, use the setDiagPlan() method and try again.

### Querying the diagnostic plan

To see the disgnostic plan that is currently set, run following wsadmin command with the jython script queryDiagPlanDmgr.py.

If the sample diagnostic plan is set, something similar to below is printed.

[root@myserver bin]# ./wsadmin.sh -username Administrator -password password -lang jython -f queryDiagPlanDmgr.py

WASX7209I: Connected to process "dmgr" on node htakamiyDmgrCellManager01 using SOAP connector; The type of process is: DeploymentManager

The following is the configured diagnostic plan:

The asterisk (\*) in front of the action indicates the currently running action.

\*MATCH: [

TRACE: CWPKI0059I\*

]

SET\_TRACESPEC: \*=info:com.ibm.ws.security.core.distSecurityComponentImpl=all:SSL=all:com.ibm.ws.management.repository.FileDocument=all:com.ibm.ws.management.FileRepository=all:com.ibm.ws.management.connector.soap.SOAPServer=all

MATCH: [

TRACE: CWPKI0060I\*

]

DELAY: 300

SET\_TRACESPEC: \*=info

DUMPBUFFER

(Repeat above for a couple of times)

[root@myserver bin]#

What above plan does is to look for “CWPKI0059I: The WSScheduler alarm started.” .When it is found, enable the trace when the certificate expiration monitor is started.

Then it looks for following message to find the schedulered task ended.

“CWPKI0060I: The WSScheduler alarm ended.”

The sample diagnostic plan waits for 5 mins (DELAY 300) to make sure remaining activities is captured in the trace then disabling the trace option. Finally the trace output in the memory is dumped out to a file: /{was\_install\_dir}/profiles/{profileName}/traceDump\_xx.xx.xx\_xx.xx.xx.log

# Collecting data

Use collector toolf[[5]](#footnote-5) to collect all the files. The diagnostic plan creates files in the directory[[6]](#footnote-6) below.

/{was\_install\_dir}/profiles/{profileName}/traceDump\_xx.xx.xx\_xx.xx.xx.log

### Reading the startup trace

The trace output of interest at server startup is between CWPKI0057I to CWPKI0058I. It shows whether the certificate expiration monitor configuration is read from security.xml successfully. It also shows the current time and the calculation of the next startup time.

Note: If the certificate expiration monitor’s scheduled time is in the past, the server clculates the next time.

Following are the trace output of our interest. It should show only one alarm in the Alarm Map and scheduled for reasonable/expected kickoff date.

CWPKI0057I: Initializing WSScheduler.

CWPKI0058I: WSScheduler has been initialized.

Trace snippet :

[7/14/19 1:30:16:254 EDT] 00000001 WSScheduler I **CWPKI0057I: Initializing WSScheduler.**

[7/14/19 1:30:16:254 EDT] 00000001 WSScheduler 3 WSSchedulerInit(SecurityConfigObject, AlarmListener,key)

[7/14/19 1:30:16:255 EDT] 00000001 WSScheduler 3 uuid: null

[7/14/19 1:30:16:255 EDT] 00000001 WSScheduler 3 Next alarm scheduled for: Mon Jul 15 00:01:47 EDT 2019

[7/14/19 1:30:16:256 EDT] 00000001 WSScheduler > setAlarm key:CertExpMonitor\_admin Alarm:\_Alarm(81092646,com.ibm.ws.crypto.config.WSScheduler@d7a7a005,java.lang.String@9463d2bb)Next Start:1563163307866Entry

[7/14/19 1:30:16:256 EDT] 00000001 WSScheduler > cancelAlarm key:CertExpMonitor\_admin Entry

[7/14/19 1:30:16:256 EDT] 00000001 WSScheduler < cancelAlarm Exit

[7/14/19 1:30:16:256 EDT] 00000001 WSScheduler 3 Next Alarm is set to 1563163307866 Mon Jul 15 00:01:47 EDT 2019

[7/14/19 1:30:16:256 EDT] 00000001 WSScheduler < setAlarm Exit

[7/14/19 1:30:16:256 EDT] 00000001 WSScheduler 3 **-- printing Alarm Map ---**

**[7/14/19 1:30:16:256 EDT] 00000001 WSScheduler 3 key=CertExpMonitor\_admin nextStartLong=1563163307866 nextStartDate=Mon Jul 15 00:01:47 EDT 2019**

**[7/14/19 1:30:16:257 EDT] 00000001 WSScheduler 3 -------------------------**

[7/14/19 1:30:16:257 EDT] 00000001 WSScheduler I **CWPKI0058I: WSScheduler has been initialized.**

# Newly added messages

Additional messages are listed below.

SECJ0446I: An SSL configuration change was found. The SSL configuration is reinitialized: {0}

explanation=The server refreshes the SSL configuration that is stored in memory.

useraction=No action is required.

SECJ0447I: An SSL configuration change was not found: {0}

explanation=The server do not refresh the SSL configuration that is stored in memory.

useraction=No action is required.

CWPKI0763I: The certificate expiration monitor started.

explanation=The certificate expiration monitor started as scheduled.

useraction=No action is required.

CWPKI0764I: The SSL configuration is refreshed when the certificate expiration monitor finishes.

explanation= The SSL configuration refreshes when the certificate expiration monitor finishes

useraction=No action is required.

CWPKI0765I: The certificate expiration monitor saved the workspace.

explanation=The changes made by the certificate expiration monitor are saved to the configuration workspace.

useraction=No action is required.

CWPKI0766I: The certificate expiration monitor finished successfully.

explanation=If the certificate is updated, the SSL configuration refreshes.

useraction=No action is required.

CWPKI0054I: The SSL configuration changed and the {0} listener is notified. The SSL configuration alias is {1}.

explanation=The component that subscribes to the SSL configuration changes is notified.

useraction=No user action is required.

CWPKI0055I: The SSL configuration is initializing.

explanation=The SSL configuration is refreshed when initialization completes.

useraction=No user action is required.

CWPKI0056I: The SSL configuration was initialized.

explanation=The SSL configuration is refreshed.

useraction=No user action is required.

CWPKI0057I: The WSScheduler is initializing.

explanation=The scheduler for the security component is initializing.

useraction=No user action is required.

CWPKI0058I: The WSScheduler was initialized.

explanation=The scheduler read the security configuration and is ready to start scheduled tasks.

useraction=No user action is required.

CWPKI0059I: The WSScheduler alarm started.

explanation=The scheduler is starting a scheduled task.

useraction=No user action is required.

CWPKI0060I: The WSScheduler alarm ended.

explanation=The scheduled task is finished.

useraction=No user action is required.

1. <https://www-01.ibm.com/support/docview.wss?uid=swg1PH06642> [↑](#footnote-ref-1)
2. <https://www-01.ibm.com/support/docview.wss?uid=swg1PI97338> [↑](#footnote-ref-2)
3. <https://www.ibm.com/support/knowledgecenter/SSAW57_9.0.5/com.ibm.websphere.nd.multiplatform.doc/ae/ttrb_diagplan.html?view=embed> [↑](#footnote-ref-3)
4. <https://www.ibm.com/support/knowledgecenter/SSAW57_9.0.5/com.ibm.websphere.nd.multiplatform.doc/ae/rtrb_diagplan.html?view=embed&origURL=SSAW57_9.0.0/com.ibm.websphere.nd.multiplatform.doc/ae/rtrb_diagplan.html> [↑](#footnote-ref-4)
5. <https://www.ibm.com/support/knowledgecenter/en/SSEQTP_8.5.5/com.ibm.websphere.base.doc/ae/ttrb_runct.html> [↑](#footnote-ref-5)
6. <https://github.ibm.com/was-liberty/WS-CD-Open/issues/17566> [↑](#footnote-ref-6)