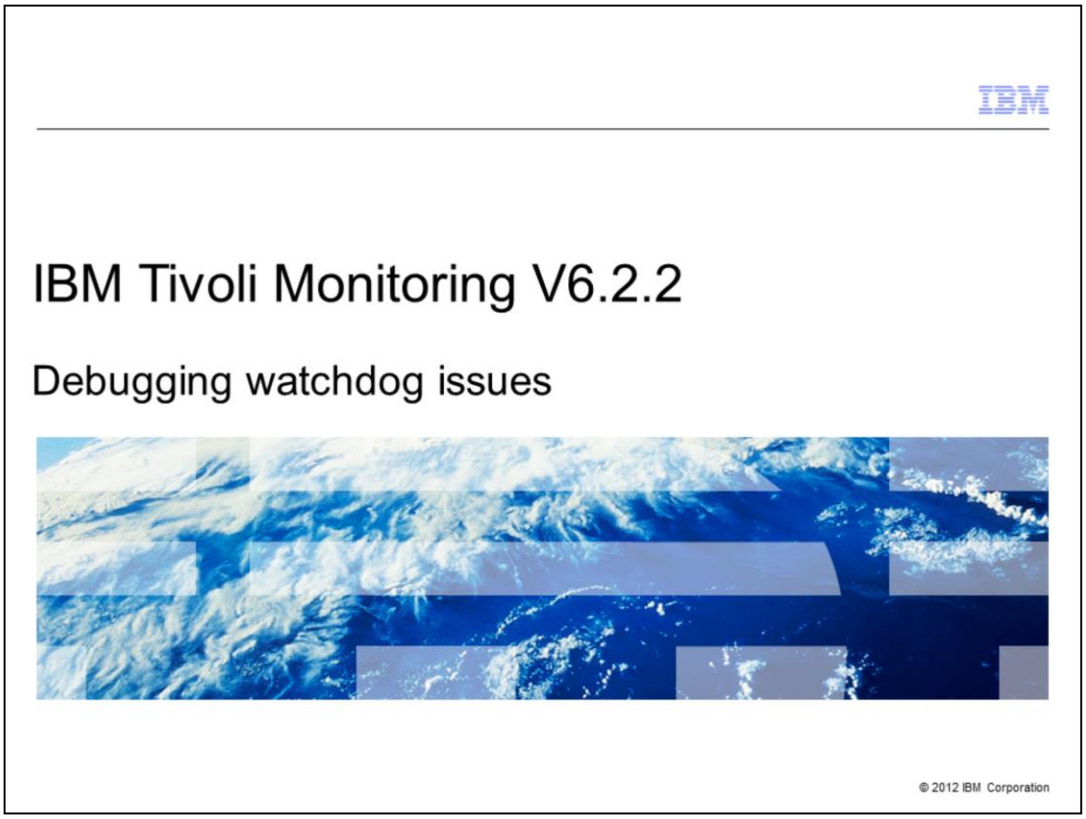
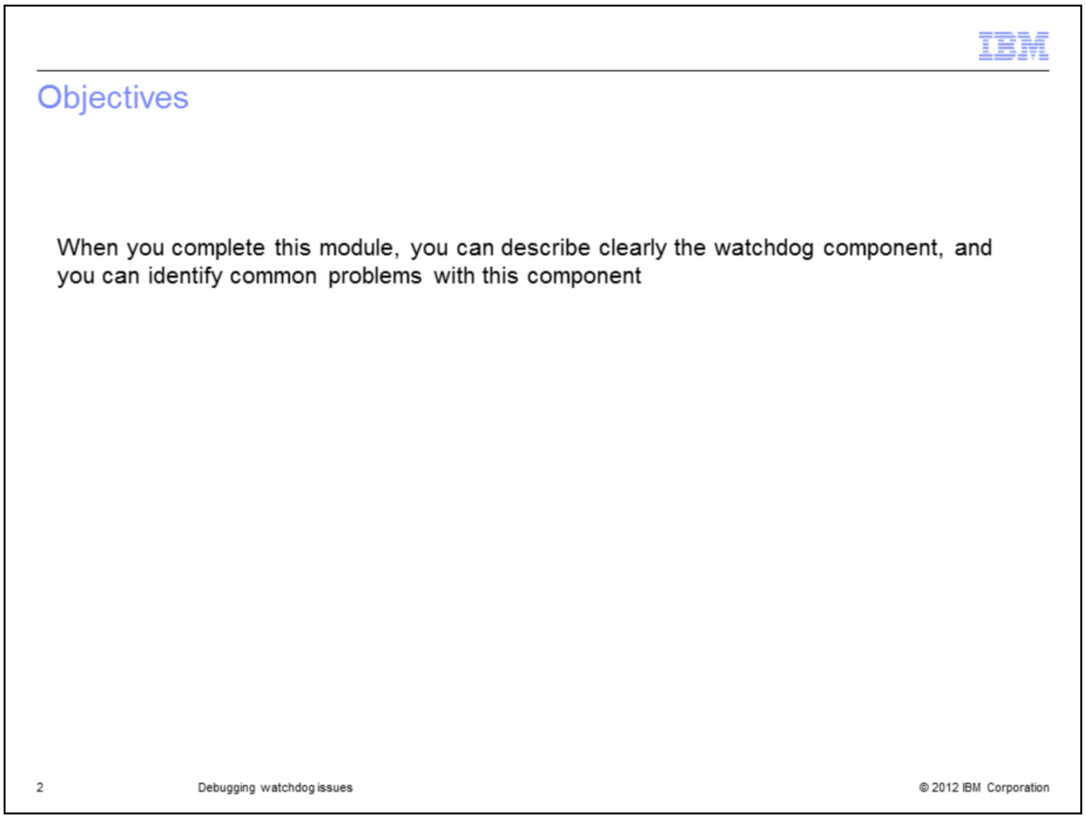


IBM Tivoli Monitoring V6.2.2

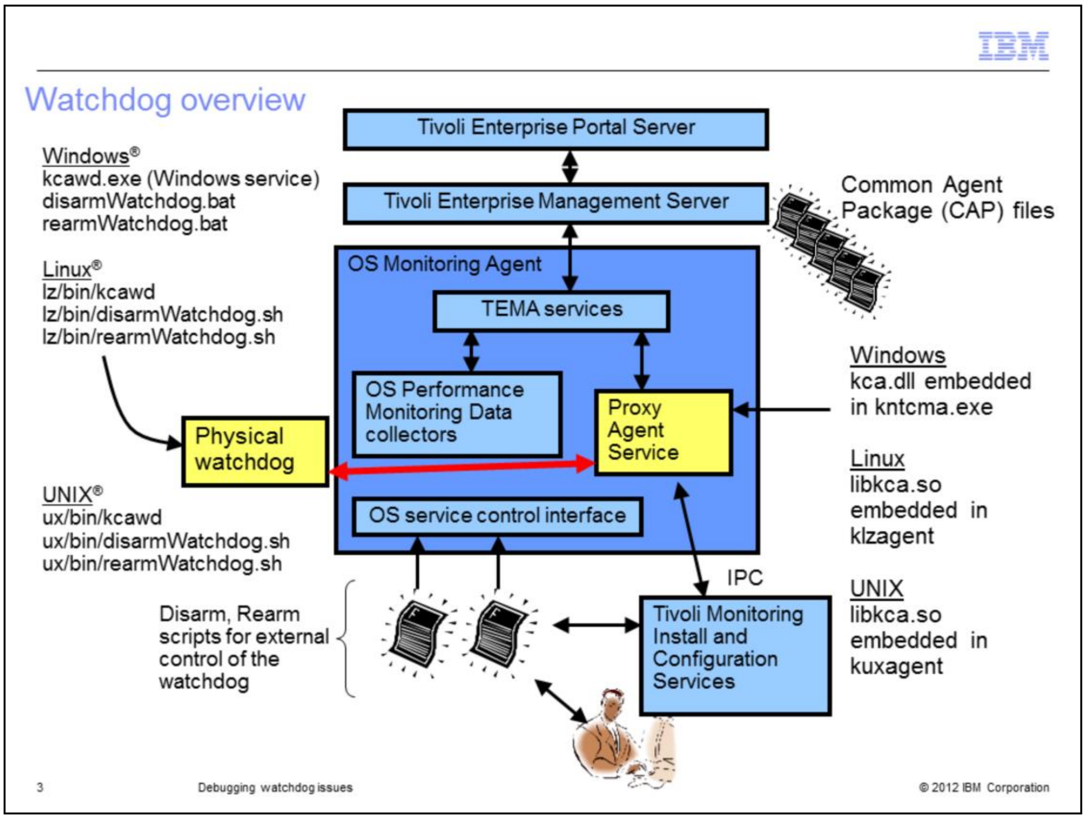
Debugging watchdog issues

IBM Tivoli® monitoring V6.2.2, debugging watchdog issues.


Objectives

When you complete this module, you can describe clearly the watchdog component, and you can identify common problems with this component

With this module, you learn about the watchdog component and how to verify whether the watchdog is the root cause of issues that you face. You can also identify some common problem scenarios with this component.



Watchdog overview

Tivoli Enterprise Portal Server
Tivoli Enterprise Management Server
Common Agent Package (CAP) files
Windows®
kcawd.exe (Windows service)
disarmWatchdog.bat
rearmWatchdog.bat
Linux®
lz/bin/kcawd
lz/bin/disarmWatchdog.sh
lz/bin/rearmWatchdog.sh
OS Monitoring Agent
TEMA services
OS Performance Monitoring Data collectors
Proxy Agent Service
Physical watchdog
Windows
kca.dll embedded in kntcma.exe
Linux
libkca.so embedded in klzagent
UNIX®
ux/bin/kcawd
ux/bin/disarmWatchdog.sh
ux/bin/rearmWatchdog.sh
OS service control interface
IPC
UNIX
libkca.so embedded in kuxagent
Disarm, Rearm scripts for external control of the watchdog
Tivoli Monitoring Install and Configuration Services

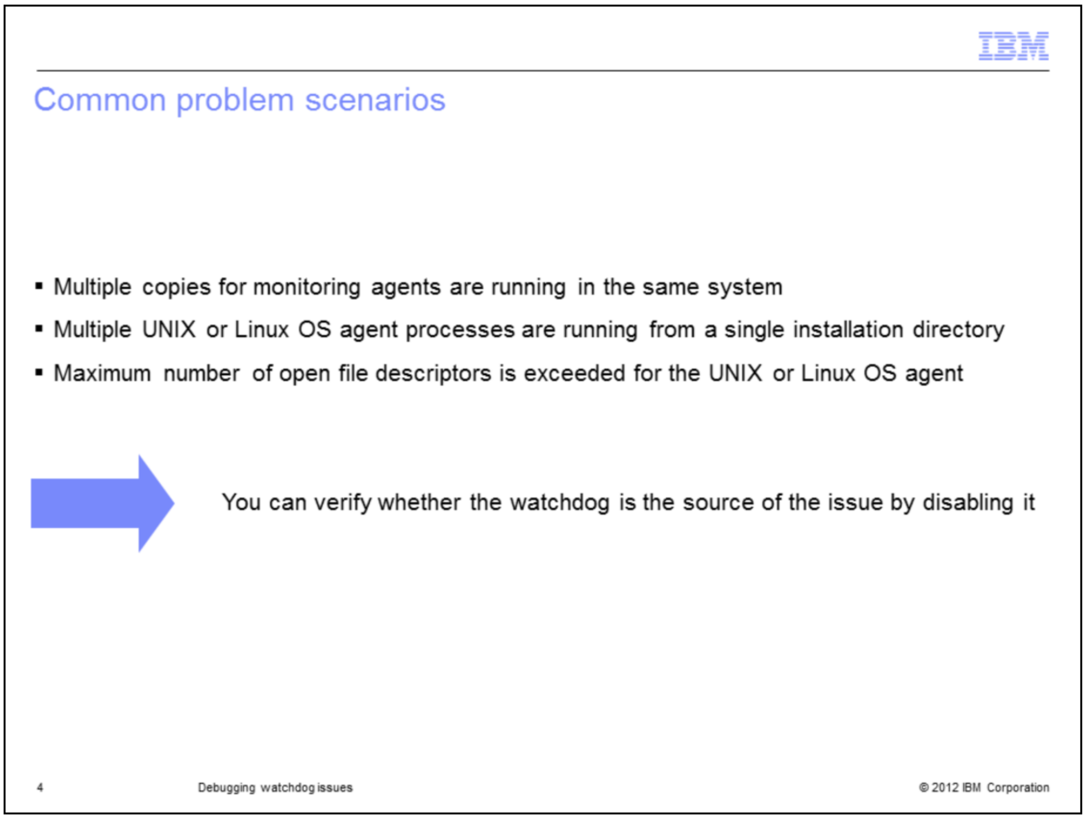
This slide provides an overview of the watchdog component. When you talk about watchdog, you have to distinguish between the Proxy Agent Service and the physical watchdog, that are shown in yellow boxes.

The Proxy Agent Service is embedded in the OS agent process and it monitors all the monitoring agents. For Windows, the **kca.dll** is embedded in **kntcma.exe**, while, for Linux and UNIX the library is **libkca.so**.

The physical watchdog is the external process that is used to monitor the availability of the OS agent itself. This process is named **kcawd** (.exe for Windows).

You can use the **disarmWatchdog** script to locally disable the watchdog for both managed agents and the OS agent itself. To rearm the watchdog, you have two possibilities: recycle the OS monitoring agent, or invoke the **rearmWatchdog** script.

Another important key factor in watchdog behavior is the Common Agent Package file, also called CAP file. For each agent that the watchdog can manage, there is a CAP file named **<ITM product code>_default.xml**. The watchdog reads each CAP file to gather all the information that is needed to check the availability of the specific agent.



Common problem scenarios

- Multiple copies for monitoring agents are running in the same system
- Multiple UNIX or Linux OS agent processes are running from a single installation directory
- Maximum number of open file descriptors is exceeded for the UNIX or Linux OS agent

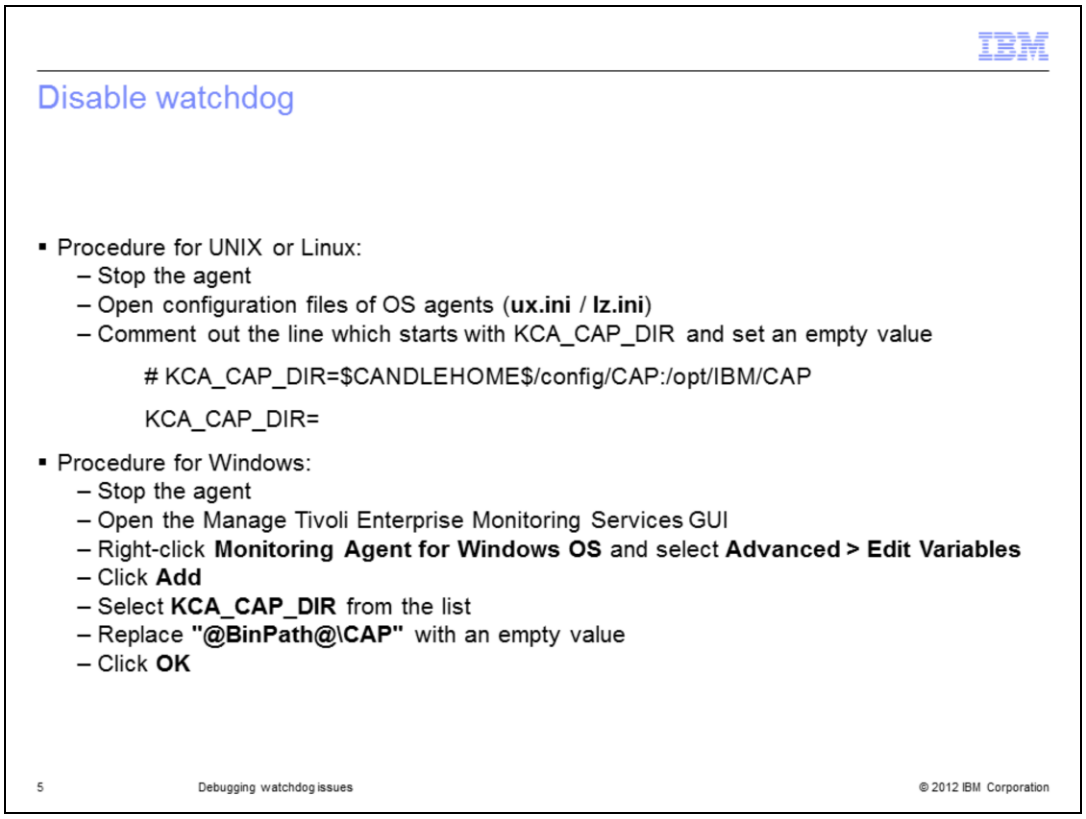
You can verify whether the watchdog is the source of the issue by disabling it

4 Debugging watchdog issues © 2012 IBM Corporation

The Watchdog feature improved the reliability and availability of IBM Tivoli Monitoring processes, restarting them in case of failure.

Sometimes Watchdog is identified as the root cause for some specific scenarios. For example, you might find multiple copies for the monitoring agent process in the system, or multiple UNIX or Linux operating system agent processes in the same installation folder. You might also see the operating system agent reaches the maximum number of open file descriptors.

In this scenario and with other issues that might occur when the operating system agent and the watchdog are running, you can identify if the watchdog is the source of the issue or not by disabling it. The next slide describes the steps to follow to disable the watchdog component.



Disable watchdog

- Procedure for UNIX or Linux:
  - Stop the agent
  - Open configuration files of OS agents (**ux.ini** / **lz.ini**)
  - Comment out the line which starts with KCA_CAP_DIR and set an empty value

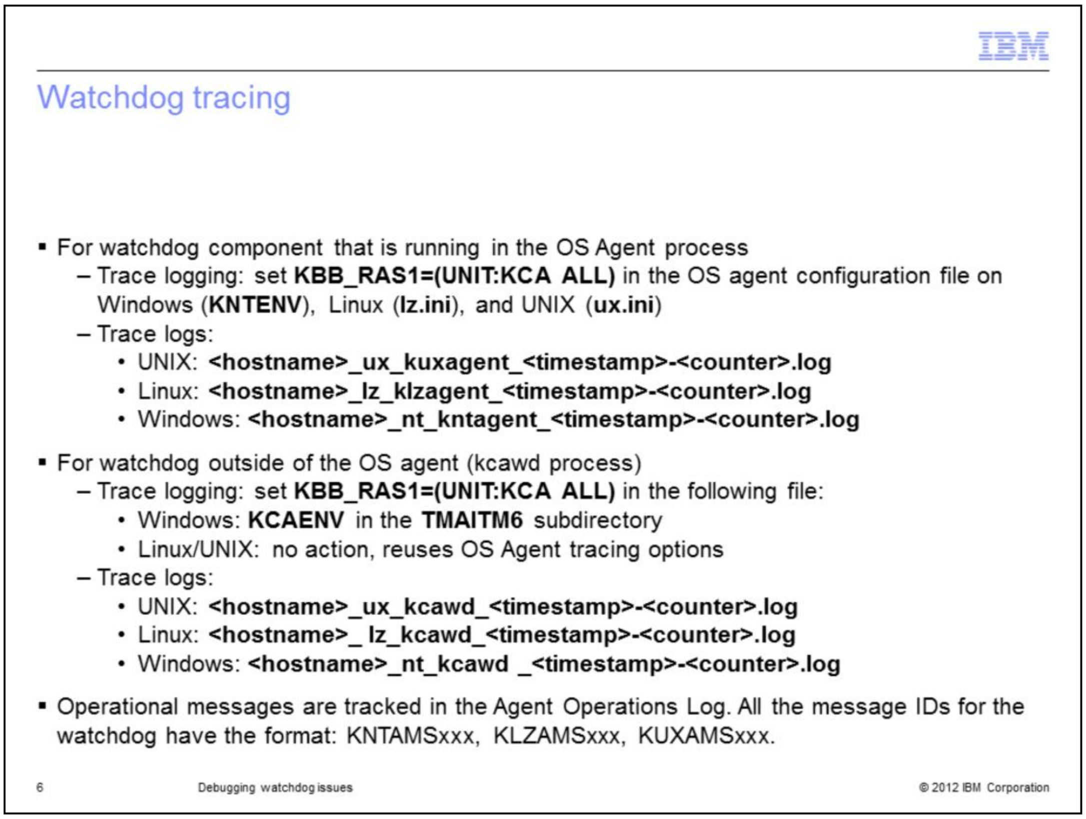
```
# KCA_CAP_DIR=$CANDLEHOME$/config/CAP:/opt/IBM/CAP
KCA_CAP_DIR=
```

- Procedure for Windows:
  - Stop the agent
  - Open the Manage Tivoli Enterprise Monitoring Services GUI
  - Right-click **Monitoring Agent for Windows OS** and select **Advanced > Edit Variables**
  - Click **Add**
  - Select **KCA_CAP_DIR** from the list
  - Replace **"@BinPath@\CAP"** with an empty value
  - Click **OK**

5 Debugging watchdog issues © 2012 IBM Corporation

Here you see the steps to disable the watchdog. When you run this procedure, the watchdog does not start again even if you restart the operating system agent. This approach is different in respect to the **disarmwatchdog** script. In fact, this script disables the watchdog temporary.

In the Windows, UNIX, or Linux cases, you have to specify an empty string for the **KCA_CAP_DIR** directory. This variable specifies the directory where the watchdog can find the configuration files that are named CAP files. If you set this variable to an empty directory, then the watchdog cannot find any configuration files and it does not start.

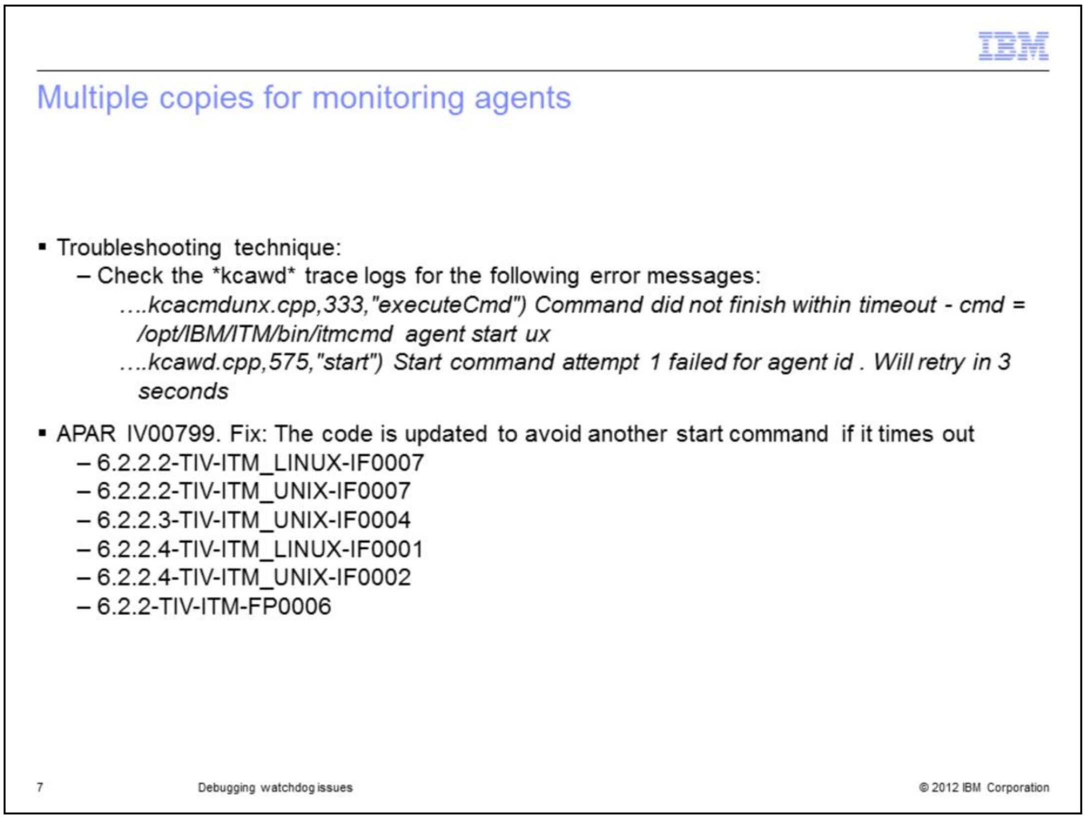


Watchdog tracing

- For watchdog component that is running in the OS Agent process
  - Trace logging: set **KBB_RAS1=(UNIT:KCA ALL)** in the OS agent configuration file on Windows (**KNTENV**), Linux (**lz.ini**), and UNIX (**ux.ini**)
  - Trace logs:
    - UNIX: **<hostname>_ux_kuxagent_<timestamp>-<counter>.log**
    - Linux: **<hostname>_lz_klzagent_<timestamp>-<counter>.log**
    - Windows: **<hostname>_nt_kntagent_<timestamp>-<counter>.log**
- For watchdog outside of the OS agent (kcawd process)
  - Trace logging: set **KBB_RAS1=(UNIT:KCA ALL)** in the following file:
    - Windows: **KCAENV** in the **TMAITM6** subdirectory
    - Linux/UNIX: no action, reuses OS Agent tracing options
  - Trace logs:
    - UNIX: **<hostname>_ux_kcawd_<timestamp>-<counter>.log**
    - Linux: **<hostname>_ lz_kcawd_<timestamp>-<counter>.log**
    - Windows: **<hostname>_nt_kcawd _<timestamp>-<counter>.log**
- Operational messages are tracked in the Agent Operations Log. All the message IDs for the watchdog have the format: KNTAMSxxx, KLZAMSxxx, KUXAMSxxx.

6 Debugging watchdog issues © 2012 IBM Corporation

Here you can find a reference for the naming convention for the watchdog trace logs. As described previously, there is the external watchdog that is named **kcawd**. The log files that contain the string 'kcawd' in the name, see the external watchdog. If you want to check the messages for the internal watchdog, you have to review the operating system agent trace logs. In these logs, you have to check for the 'kca' modules.



Multiple copies for monitoring agents

- Troubleshooting technique:
  - Check the *kcawd* trace logs for the following error messages:

```
….kcacmdunx.cpp,333,"executeCmd") Command did not finish within timeout - cmd =
  /opt/IBM/ITM/bin/itmcmd agent start ux
….kcawd.cpp,575,"start") Start command attempt 1 failed for agent id . Will retry in 3
  seconds
```

- APAR IV00799. Fix: The code is updated to avoid another start command if it times out
  - 6.2.2.2-TIV-ITM_LINUX-IF0007
  - 6.2.2.2-TIV-ITM_UNIX-IF0007
  - 6.2.2.3-TIV-ITM_UNIX-IF0004
  - 6.2.2.4-TIV-ITM_LINUX-IF0001
  - 6.2.2.4-TIV-ITM_UNIX-IF0002
  - 6.2.2-TIV-ITM-FP0006

7 Debugging watchdog issues © 2012 IBM Corporation

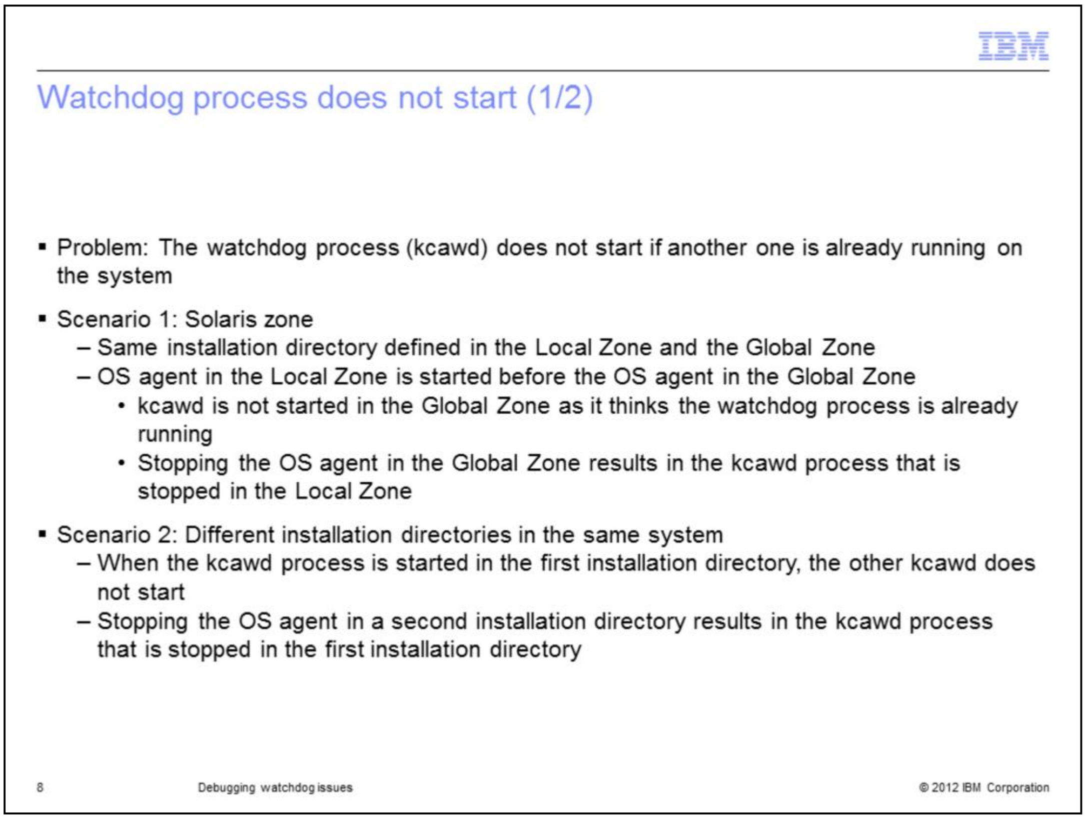
Here you can see some common problem scenarios in which the watchdog component is involved.

In this case, you are using the watchdog and, based on the **ps -ef** output, you can see that there are multiple copies for the same monitoring agents. If you disable the watchdog, then the issue does not occur. In this way, you are sure that the watchdog is causing this problem. As a second step, check the **kcawd** trace logs. In this scenario, you can find error messages such as the ones that are shown in this slide.

These messages describe a timeout message and then a start command. This condition can happen when the watchdog restarts an agent that it determines is down, and the start command, **itmcmd**, does not complete within the timeout. In this scenario, the **itmcmd** is left to finish and another start command is started.

This condition can cause multiple copies of the same monitoring agent.

This problem is addressed in APAR IV00799, and here you can find the list of the fixes.


Watchdog process does not start (1/2)

- Problem: The watchdog process (kcawd) does not start if another one is already running on the system
- Scenario 1: Solaris zone
  - Same installation directory defined in the Local Zone and the Global Zone
  - OS agent in the Local Zone is started before the OS agent in the Global Zone
    - kcawd is not started in the Global Zone as it thinks the watchdog process is already running
    - Stopping the OS agent in the Global Zone results in the kcawd process that is stopped in the Local Zone
- Scenario 2: Different installation directories in the same system
  - When the kcawd process is started in the first installation directory, the other kcawd does not start
  - Stopping the OS agent in a second installation directory results in the kcawd process that is stopped in the first installation directory

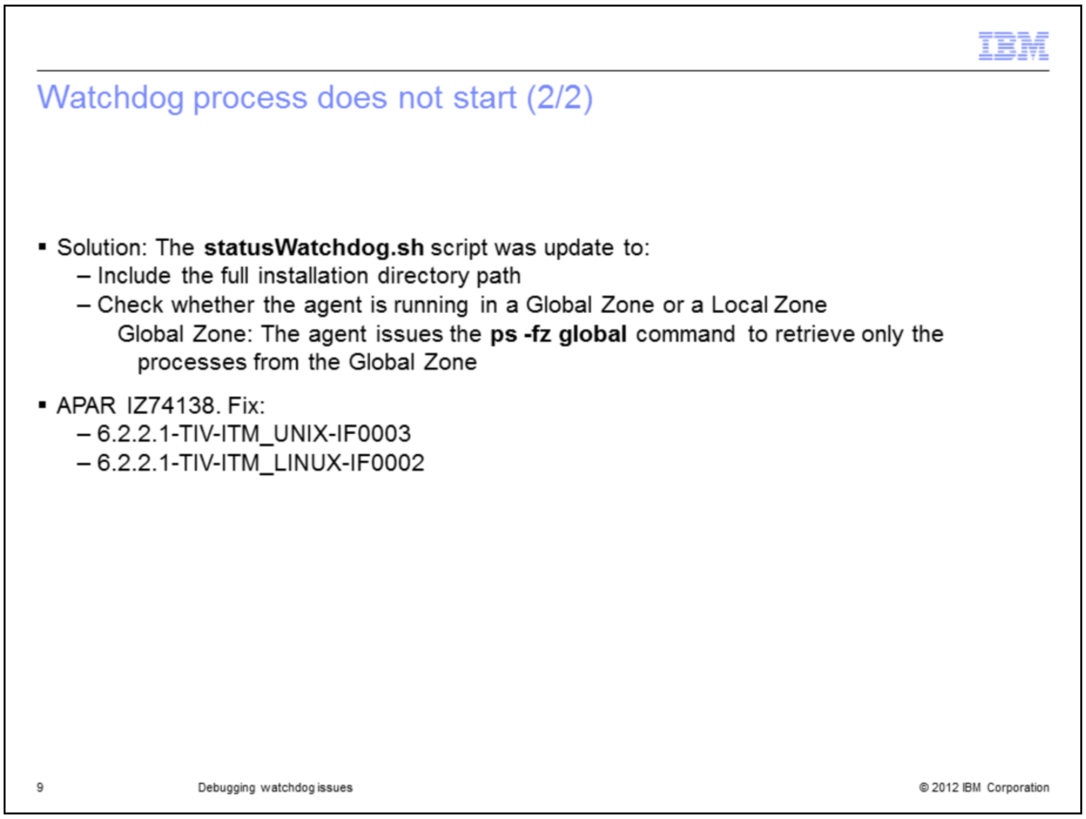
8 Debugging watchdog issues © 2012 IBM Corporation

In this common problem with the watchdog, you cannot start the **kcawd** process if another one is already running in the system.

If you are facing this issue, you have to verify which of two scenarios you are in.

The first one: you installed the operating system agent in the Local and Global Zones of your Solaris zone box. You started the agent in the Local Zone and then the one in the Global Zone. In this condition, the watchdog in the Global Zone does not start because another **kcawd** process is already running. Besides, if you stop the operating system agent in the Global Zone, the watchdog in the Local Zone is also stopped.

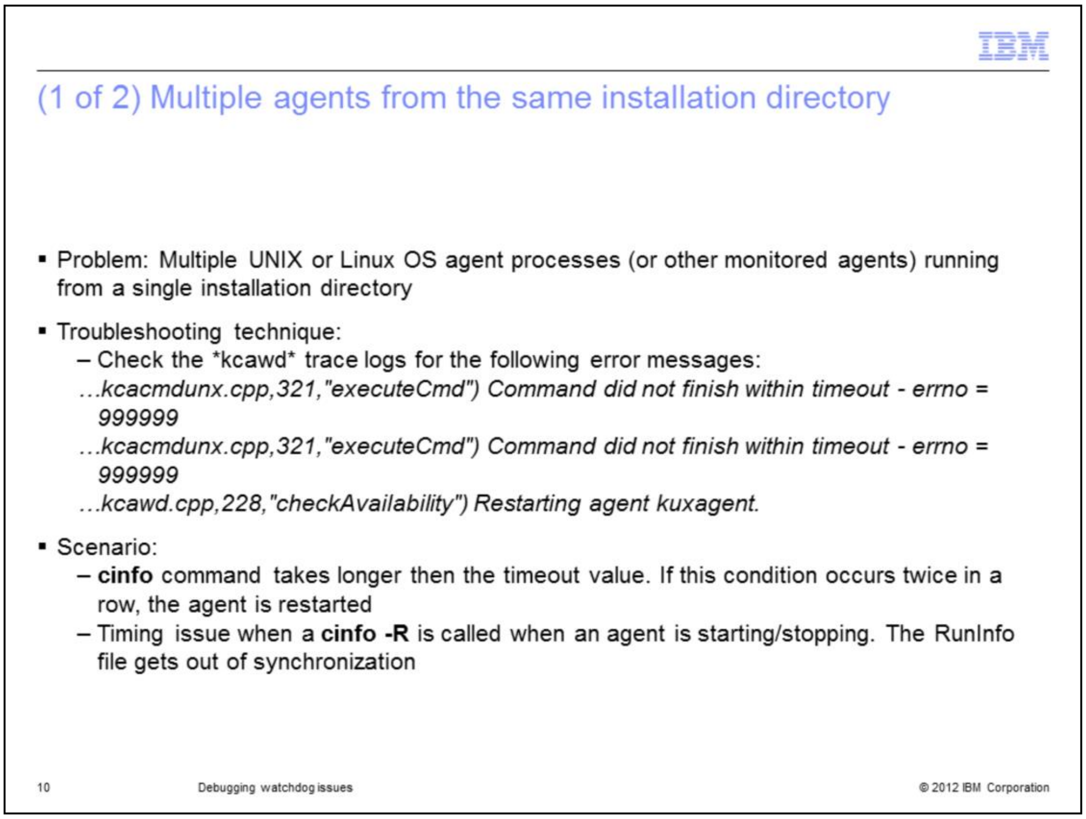
This problem happens if you are using a Solaris machine, but you can face a similar issue if you have two different installation directories in the same system. As for the Solaris case, when the watchdog is started in the first monitoring home, the second **kcawd** process cannot start. If you stop the operating system agent in the second installation directory, then the watchdog in the first monitoring home is stopped.


Watchdog process does not start (2/2)

- Solution: The **statusWatchdog.sh** script was update to:
  – Include the full installation directory path
  – Check whether the agent is running in a Global Zone or a Local Zone
    Global Zone: The agent issues the **ps -fz global** command to retrieve only the processes from the Global Zone
- APAR IZ74138. Fix:
  – 6.2.2.1-TIV-ITM_UNIX-IF0003
  – 6.2.2.1-TIV-ITM_LINUX-IF0002

This problem happens because the watchdog does not take into account the installation directory and the Solaris zone information. APAR IZ74138 addresses this problem. With this fix, the **statusWatchdog** script includes the full path of the installation directory. It also checks whether the agent is running on a Local or Global Zone by leveraging on the **ps -fz global** command to retrieve data for the Global Zone only.

Here you can find the fixes that contain this APAR.



(1 of 2) Multiple agents from the same installation directory

- Problem: Multiple UNIX or Linux OS agent processes (or other monitored agents) running from a single installation directory
- Troubleshooting technique:
  - Check the *kcawd* trace logs for the following error messages:

```
…kcacmdunx.cpp,321,"executeCmd") Command did not finish within timeout - errno = 999999
…kcacmdunx.cpp,321,"executeCmd") Command did not finish within timeout - errno = 999999
…kcawd.cpp,228,"checkAvailability") Restarting agent kuxagent.
```

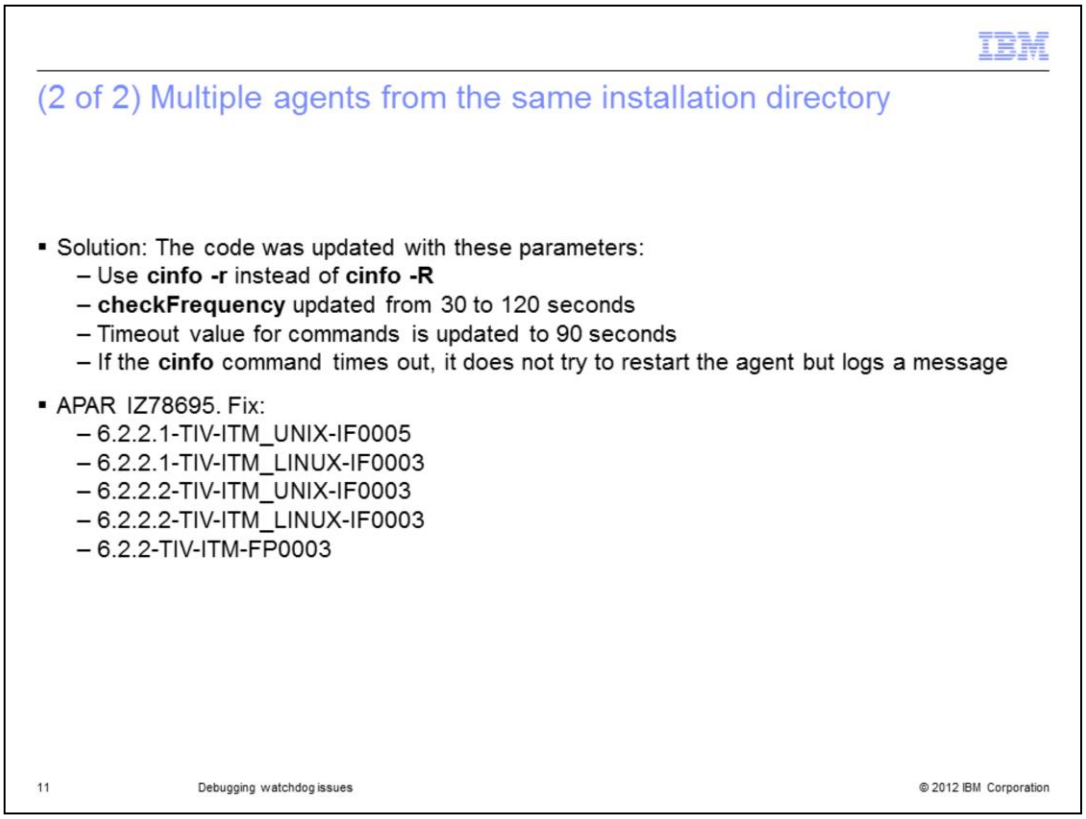
- Scenario:
  - **cinfo** command takes longer then the timeout value. If this condition occurs twice in a row, the agent is restarted
  - Timing issue when a **cinfo -R** is called when an agent is starting/stopping. The RunInfo file gets out of synchronization

10 Debugging watchdog issues © 2012 IBM Corporation

In this problem scenario, you are using the watchdog and, based on the **ps -ef** output, you can see that there are multiple copies for the monitoring agents in the same installation directory. If you disable the watchdog, then the issue does not occur. In this way, you are sure that the watchdog is causing this problem. As a second step, check the **kcawd** trace logs to find the same error messages as the ones that are shown in this slide.

These messages describe that the watchdog runs a command that does not complete within a timeout value. As a consequence, the watchdog starts the OS agent process again.

This issue can happen when the **cinfo** command takes too much time to complete and exceed the timeout value.


(2 of 2) Multiple agents from the same installation directory

- Solution: The code was updated with these parameters:
  - Use **cinfo -r** instead of **cinfo -R**
  - **checkFrequency** updated from 30 to 120 seconds
  - Timeout value for commands is updated to 90 seconds
  - If the **cinfo** command times out, it does not try to restart the agent but logs a message
- APAR IZ78695. Fix:
  - 6.2.2.1-TIV-ITM_UNIX-IF0005
  - 6.2.2.1-TIV-ITM_LINUX-IF0003
  - 6.2.2.2-TIV-ITM_UNIX-IF0003
  - 6.2.2.2-TIV-ITM_LINUX-IF0003
  - 6.2.2-TIV-ITM-FP0003

11 Debugging watchdog issues © 2012 IBM Corporation

The code of the watchdog component was updated to avoid this problematic situation by changing a few parameters.

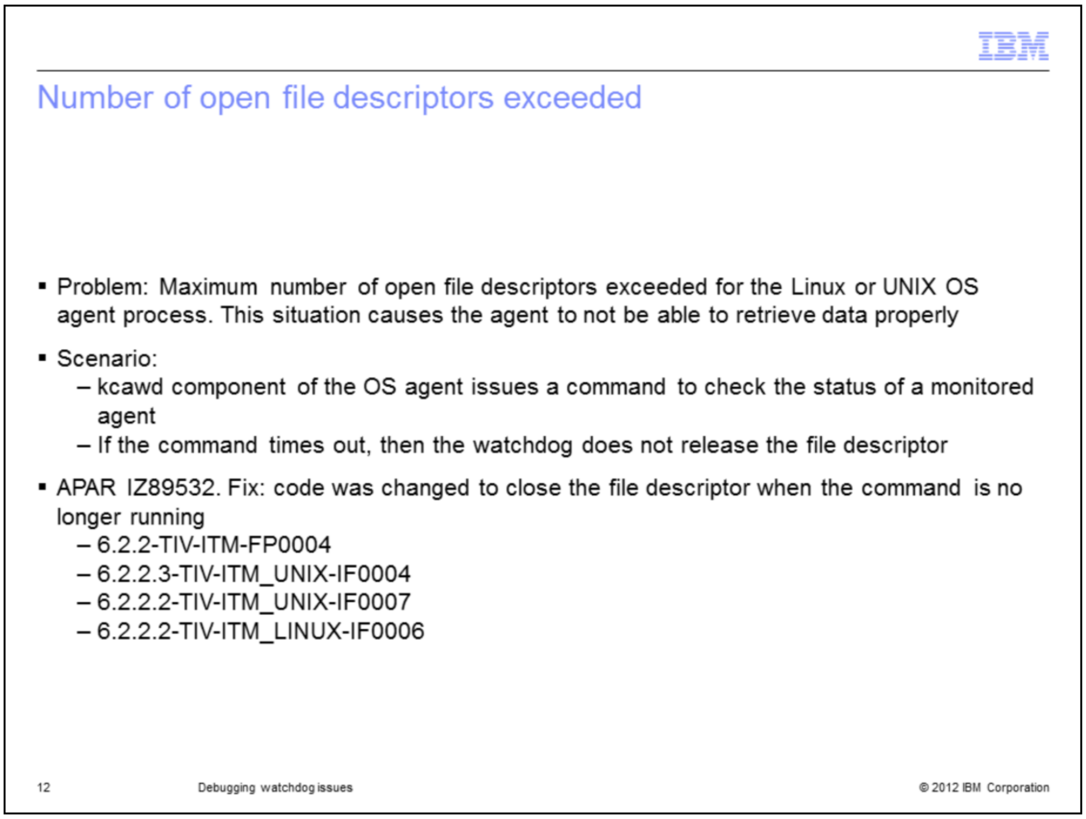
- Using the **cinfo -r** command instead of the **cinfo -R** command.

- Increasing the **checkFrequency** value from **30** to **120** seconds. It represents the length of time between availability checks of the watchdog.

- Increasing the timeout value.

- Avoiding a restart the monitoring agent if the **cinfo** command times out. …

These code changes come with APAR IZ78695. Here you can find the list of the fixes.



Number of open file descriptors exceeded

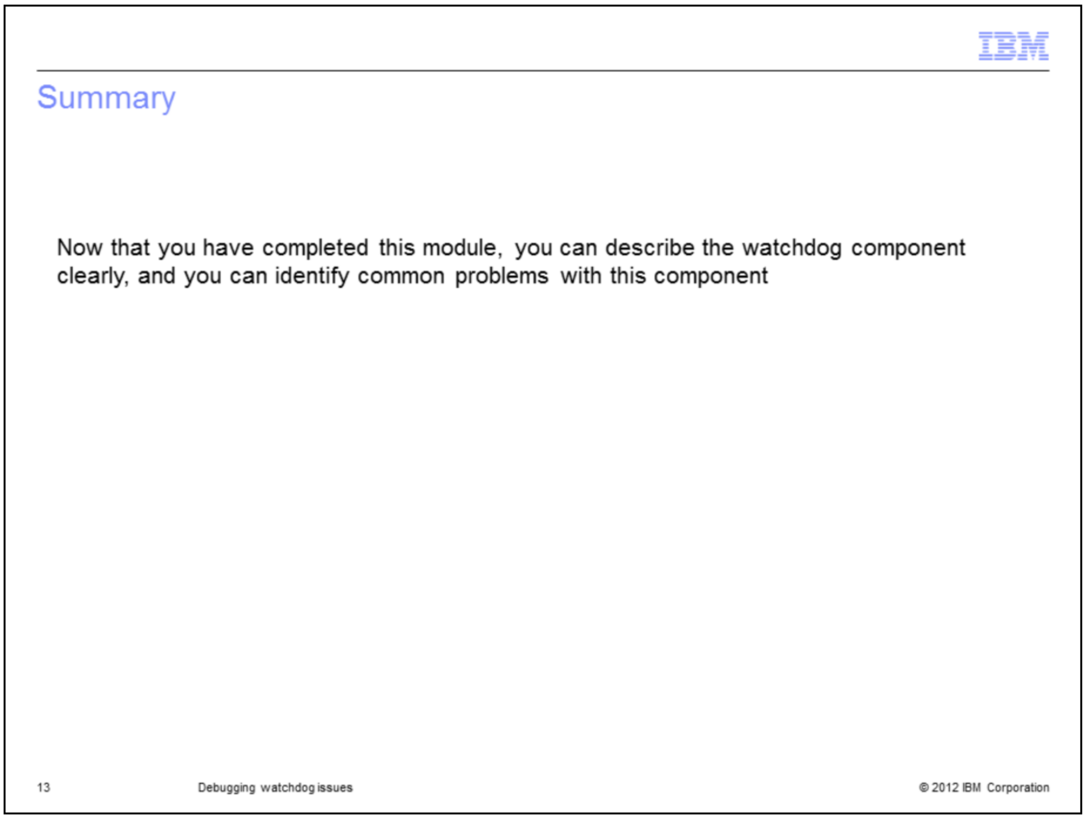
- Problem: Maximum number of open file descriptors exceeded for the Linux or UNIX OS agent process. This situation causes the agent to not be able to retrieve data properly
- Scenario:
  - kcawd component of the OS agent issues a command to check the status of a monitored agent
  - If the command times out, then the watchdog does not release the file descriptor
- APAR IZ89532. Fix: code was changed to close the file descriptor when the command is no longer running
  - 6.2.2-TIV-ITM-FP0004
  - 6.2.2.3-TIV-ITM_UNIX-IF0004
  - 6.2.2.2-TIV-ITM_UNIX-IF0007
  - 6.2.2.2-TIV-ITM_LINUX-IF0006

12 Debugging watchdog issues © 2012 IBM Corporation

In this problem scenario, the Linux or UNIX agent exceeded the number of the open file descriptors. As a consequence, the agent is not able to retrieve the data properly, and you cannot see the information in the Tivoli Enterprise Portal.

If you disable the watchdog, then the issue does not occur. The watchdog is causing this issue.

APAR IZ89532 addresses this problem. Before this APAR, the watchdog did not release the file descriptor if the command run to check the status of the monitored agent timed out.

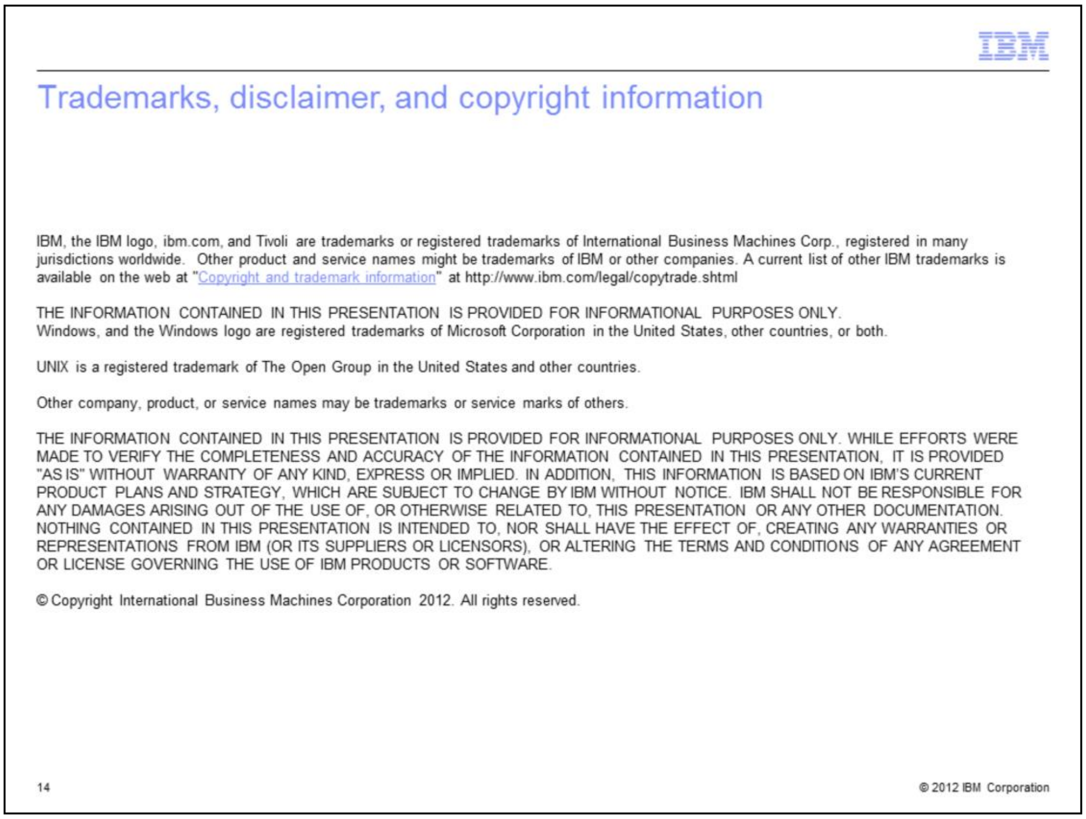


Summary

Now that you have completed this module, you can describe the watchdog component clearly, and you can identify common problems with this component

13 Debugging watchdog issues © 2012 IBM Corporation

Now you have completed this module, you can clearly describe the watchdog component and you can verify whether the watchdog is the root cause of issues that you face. You can also identify common problem scenarios with this component.



Trademarks, disclaimer, and copyright information