

Contents

- 1 Scenario Setup
- 2 Problem Statement
- 3 Chat Instability
- 4 JBoss Application Server Health
- 5 Access Log Polling
 - ◆ 5.1 Customer Session
 - ◆ 5.2 Agent Session
- 6 Customer Receiving Duplicate Agent Messages
- 7 What does this mean?
 - ◆ 7.1 What if the duplicate messages were seen in the Agent's chat window?
 - ◆ 7.2 Isolation
- 8 Web Server Issues
 - ◆ 8.1 Validation
- 9 Resolution

Scenario Setup

Web Server

1. Enabled below settings on DefaultAppPool:
 1. PeriodicRestartTime = 1740
 2. IdleTimeout = 20
 3. AppPoolQueueLength = 1000
 4. PingingEnabled = True
 5. RapidFailProtection = True

Agent PC

1. Log in as integrated agent "goofy" and go Available for Chat
2. Initiate chat from integrated customer entry point

Problem Statement

Chats are unstable, messages do not arrive promptly or session is disconnected.

Chat Instability

The root cause of chat issues can vary just like with any other issue, so a clear problem description is crucial. If there is a report of chat instability, what is really going on?

- Are customers getting disconnected?
- Are messages being received multiple times or after a long delay?
- Are agent consoles crashing?
- etc.

In this scenario, we will focus on issues with the chat session's connection. Outward behavior can include any of the following:

Chat_Session_Delays

- Customer messages being received by Agent multiple times
- Agent messages being received by Customer multiple times
- Delivery of messages between Agent/Customer being delayed or not received at all
- Agents being alerted of new chat sessions, but being disconnected for no clear reason

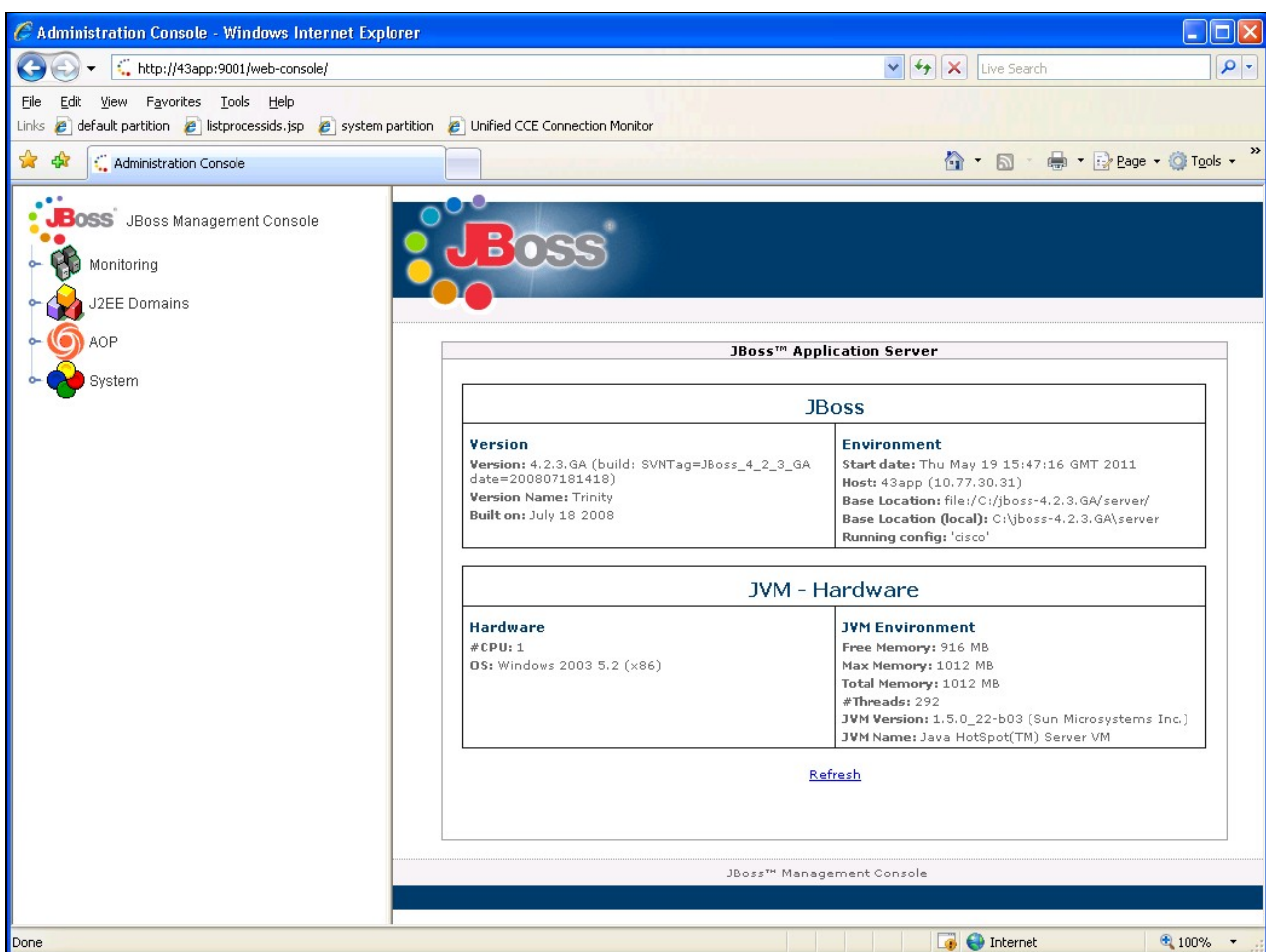
JBoss Application Server Health

The first thing to check is your Application Server. Use the below query to determine which Application Server the chat activity occurred on:

```
select activity_id, attendee_home from eglv_attendee where activity_id = <ACTIVITY_ID>
```

Then go to that Application Server's JBoss Web Console:

`http://<appserver>:9001/web-console/`



The screenshot shows the JBoss Management Console in a Windows Internet Explorer browser. The address bar shows `http://43app:9001/web-console/`. The console displays the following information:

JBoss	
Version Version: 4.2.3.GA (build: SVNTag=JBoss_4_2_3_GA date=200807181418) Version Name: Trinity Built on: July 18 2008	Environment Start date: Thu May 19 15:47:16 GMT 2011 Host: 43app (10.77.30.31) Base Location: file:/C:/jboss-4.2.3.GA/server/ Base Location (local): C:\jboss-4.2.3.GA\server Running config: 'cisco'

JVM - Hardware	
Hardware # CPU: 1 OS: Windows 2003 5.2 (x86)	JVM Environment Free Memory: 916 MB Max Memory: 1012 MB Total Memory: 1012 MB #Threads: 292 JVM Version: 1.5.0_22-b03 (Sun Microsystems Inc.) JVM Name: Java HotSpot(TM) Server VM

Below the JVM - Hardware section, there is a [Refresh](#) link.

Ensure that the system has adequate memory free, and that the #Threads is not too high. For reference, #Threads should not exceed 300 per 200 agents. In this case we are fine as our thread count is below 300.

Note that there are optimizations described in CSCt171368 that should be applied to workers.properties on the Web Server, and server.xml on the File Server to better manage JBoss thread count. These should be applied if issues are observed with JBoss stability.

Access Log Polling

The Application Server keeps track of all JSP requests and chat sessions (both Agent and Customer) in the Access Logs, located on the File Server. Think of it as similar to the IIS logs, only for the JBoss Application Server.

When a Chat session is active, both the Agent and Customer sessions poll the Application Server (through the Web Server) throughout the lifespan of the chat. These messages can all be seen by searching the Access Logs for "command=poll". In order to track a particular chat, there are key differences in the user_id that will be seen for the poll messages for Customer and Agent.

Customer Session

```
10.77.30.8 - - [01/Jun/2011:19:53:00 +0000] "GET /system/mr_pushlet.egain?user_id=1%241254cust1007
```

The Customer Session is easy to find. Its user_id is populated with the ActivityID and a unique CustomerID. From the above log snippet: user_id=1%241254cust1007

Agent Session

```
10.77.30.8 - - [01/Jun/2011:19:53:00 +0000] "GET /system/mr_pushlet.egain?user_id=1%241002&partiti
```

The Agent's poll messages can be found with the Agent's eGAgentID. In this case: user_id=1%241002. However, this does not include the activity ID, so will require some timestamp matching in log analysis with the Customer Session.

Customer Receiving Duplicate Agent Messages

We've received a report that Agent 60280 was chatting with a customer in activity ID 584807. The customer stated that they were receiving the Agent's messages twice in the chat window. Since we only see duplicate messages on the customer's side and not the agent, we will investigate the customer's session in the Access Logs.

First search the access_logs for the ActivityID to get the customerID. For ActivityID 584807, the customer's chat session can be tracked with the ID ?cust70706? in the logs.

```
10.1.244.9 - - [28/Jan/2011:21:09:29 +0000] "GET /system/mr_pushlet.egain?user_id=1%24584807cust70
```

Notice the value ?cnt=102? ? this is the polling count for the customer chat session. Think of it somewhat like a heartbeat between the customer's PC and the web server. So at this point we've received 102 polls.

The next message is 10s later (which is fine) and increments to 103:

```
10.1.244.9 - - [28/Jan/2011:21:09:39 +0000] "GET /system/mr_pushlet.egain?user_id=1%24584807cust70
```

Next one is 30s later (also fine) and increments to 104:

```
10.1.244.9 - - [28/Jan/2011:21:10:10 +0000] "GET /system/mr_pushlet.egain?user_id=1%24584807cust70
```

Then 20s later to 105:

```
10.1.244.9 - - [28/Jan/2011:21:10:30 +0000] "GET /system/mr_pushlet.egain?user_id=1%24584807cust70
```

Chat_Session_Delays

Then, we have a problem. **Polls 106 and 107 are never received by the Application Server.** We go just over 2 minutes before receiving the 108th poll:

```
10.1.244.9 - - [28/Jan/2011:21:12:31 +0000] "GET /system/mr_pushlet.egain?user_id=1%24584807cust70
```

Then another is missed - 109.

```
10.1.244.9 - - [28/Jan/2011:21:13:32 +0000] "GET /system/mr_pushlet.egain?user_id=1%24584807cust70
```

Then, cust70706 is fine for the rest of the session at which point the customer ends it:

```
10.1.244.9 - - [28/Jan/2011:21:37:48 +0000] "GET /system/LiveCustomerServlet.egain?eglvcmd=CustEnc
```

What does this mean?

There is some sort of issue between the customer's PC and the Web Server. When a customer starts a chat session a thread is opened between the customer and the webserver. When there is a poor connection, the server tries to establish a second connection. If it doesn't get a response it might send yet another thread. When one of these threads finally makes it through, all of the previous messages will come with it. This results in the repeated messages seen on the customer chat window.

What if the duplicate messages were seen in the Agent's chat window?

The same 'poor connection' root cause holds true, however this time it is between the agent and the Web Server.

Isolation

The webserver is obviously the central point here ? so when both agent and customer are seeing duplicate messages, it means one of 2 things:

1. There are network problems between customer/webserver, and agent/Web Server
2. There is an issue with the Web Server.

When the issue occurs on a regular basis, #2 is likely the culprit.

Web Server Issues

The only component used by EIM/WIM on the Web Server is Internet Information Services (IIS). Within IIS, there are settings that must be configured per the Installation Guide. These items are commonly overlooked or sometimes even reset by Domain Startup Scripts.

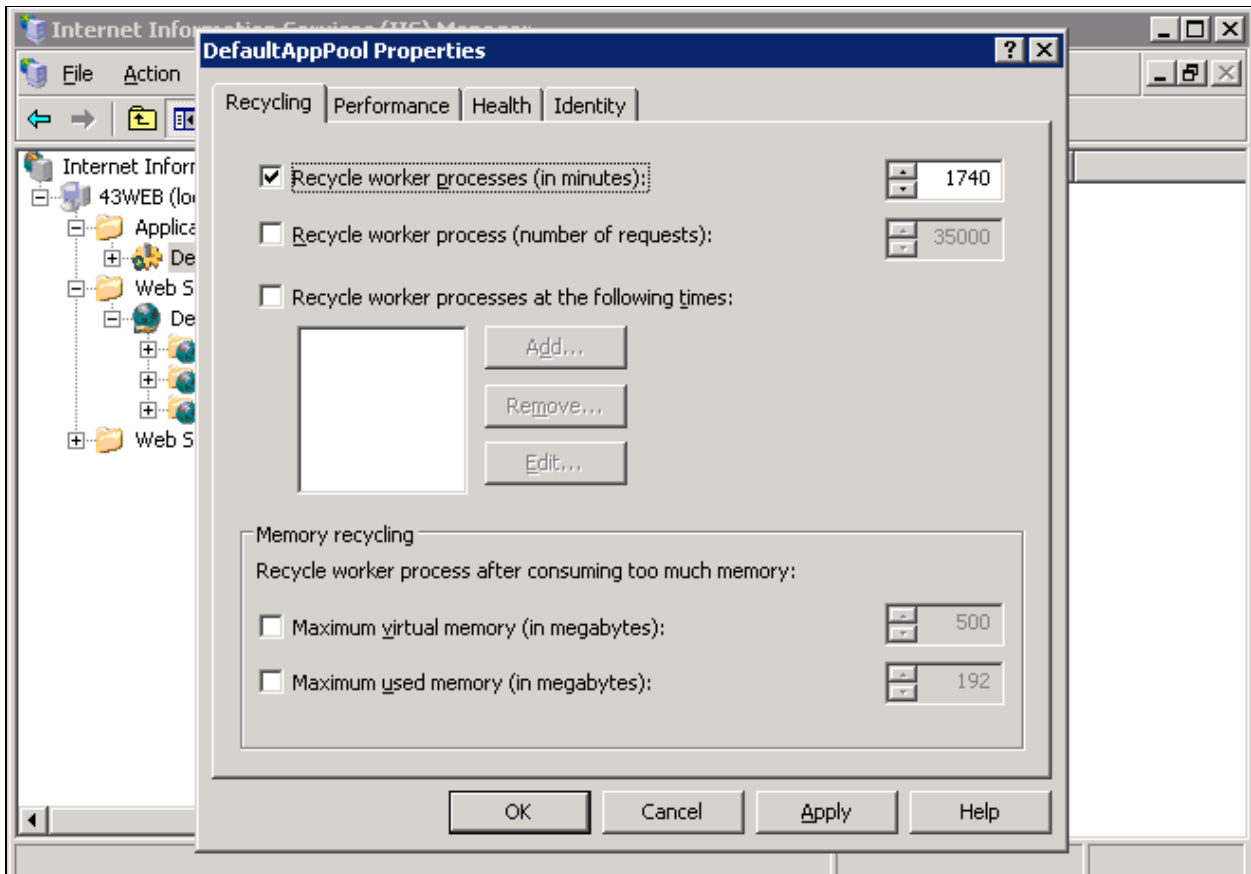
Let's validate the current settings with those located in Chapter 5 of the [4.3 Installation Guide](#), "Configuring Web Servers"

Validation

From IIS > Application Pool > DefaultAppPool > Properties, we can see several settings are incorrect.

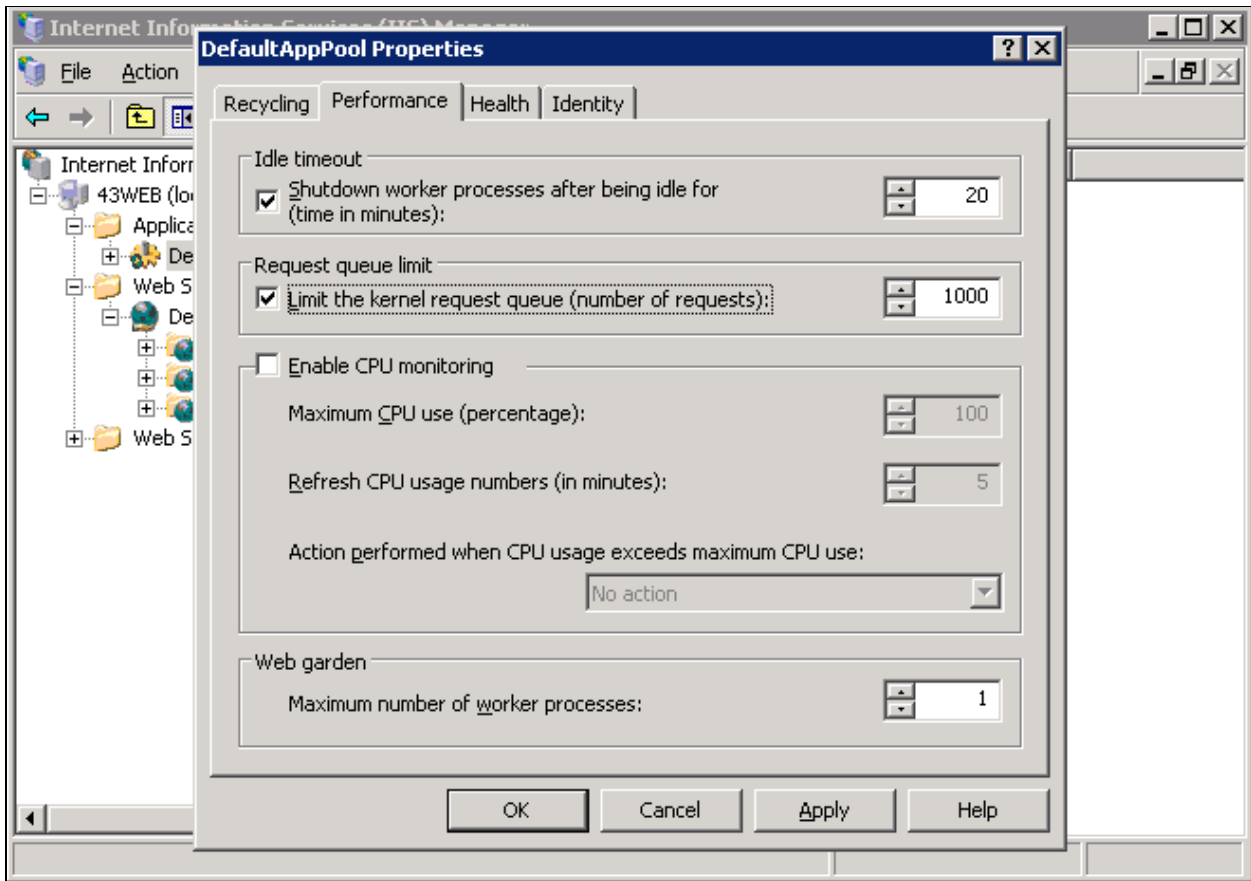
- "Recycle worker processes (in minutes):" should be DISABLED. The current setting of 1740 minutes equates to 29 hours and results in rolling interruption to service as the recycle time processes each day through the week.

Chat_Session_Delays

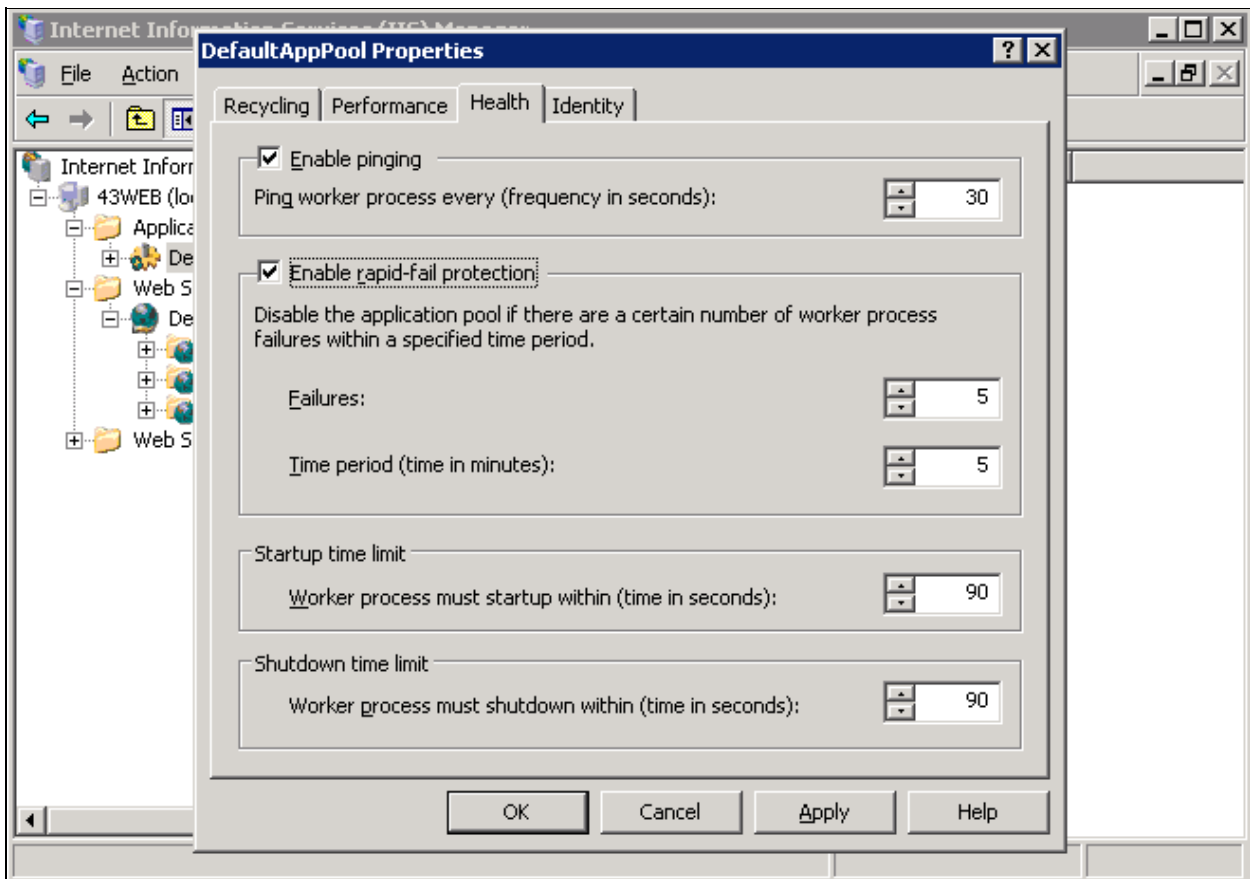


- "Idle Timeout", "Request queue limit" should be DISABLED. The current settings will recycle processes unnecessarily, resulting in interruptions to service.

Chat_Session_Delays



- "Enable ping", "Enable rapid-fail protection" should be DISABLED. Having these enabled results in interruptions to service from recoverable error conditions



Resolution

These incorrect IIS settings would certainly introduce issues in a deployment that could be very difficult to track down. They are a necessary baseline for troubleshooting any sort of stability or delay issues, especially those with Chat Sessions.