xBand xConnect Troubleshooting Guide

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev** | **Date** | **Author** | **Description** |
| 1.0 | 5/1/12 | Iwona Glabek | Release Version |

**Document Approvers & Sign-Off**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Approver** | **Role** | **Document Accept/Reject** |
| 5/1/12 | Manny Vellon | Synapse CTO | Accept |

**Table of Contents**

[1 Introduction 5](#_Toc324500966)

[1.1 Purpose 5](#_Toc324500967)

[1.2 Scope 5](#_Toc324500968)

[1.3 Assumptions and Constraints 5](#_Toc324500969)

[1.4 Definitions 5](#_Toc324500970)

[2 xBRC Theory of Operation 6](#_Toc324500971)

[2.1 xBRC Architecture Diagram 6](#_Toc324500972)

[2.2 Overall xBRC Data Flow 7](#_Toc324500973)

[3 Diagnosing xBRC Problems 7](#_Toc324500974)

[3.0 Before You Start 7](#_Toc324500975)

[3.1 Is my xBRC running? 8](#_Toc324500976)

[3.2 What to do if the xBRC is not running 8](#_Toc324500977)

[3.3 Is my xBRC “healthy”? 9](#_Toc324500978)

[3.4 How to (easily) tell if an xBRC is receiving data from its readers 10](#_Toc324500979)

[3.5 Another way to tell if an xBRC is receiving data from its readers 11](#_Toc324500980)

[3.6 Are Band IDs being properly mapped to guests? 12](#_Toc324500981)

[3.7 Is my xBRC is singulating guests? 12](#_Toc324500982)

[3.8 Is my xBRC generating messages? 13](#_Toc324500983)

[3.9 How to view xBRC messages 13](#_Toc324500984)

[3.10 How to tell whether an xBRC is configured to send messages to Sonic MQ or to an HTTP destination 14](#_Toc324500985)

[3.11 Are messages being sent upstream? 14](#_Toc324500986)

[3.12 How can I tell if an xBRC is reporting errors? 15](#_Toc324500987)

[4 Diagnosing xBR Problems 16](#_Toc324500988)

[4.1 Before You Start 16](#_Toc324500989)

[4.2 The xBRC is OK, but I still have problems 16](#_Toc324500990)

[4.3 Faulty xBR Ethernet 16](#_Toc324500991)

[4.4 The LRR Application is stopped 17](#_Toc324500992)

[5 xTP New Unit Testing 18](#_Toc324500993)

[6 xBRMS Health Page 19](#_Toc324500994)

[6.1 xBRMS Health Status Messages 20](#_Toc324500995)

[7 Diagnosing Park Entry Problems 21](#_Toc324500996)

[7.1 Park Entry XBRMS Health Status Messages 21](#_Toc324500997)

[8 Diagnosing xBRMS Problems 24](#_Toc324500998)

[8.1 Before You Start 24](#_Toc324500999)

[8.2 Is my xBRMS running? 24](#_Toc324501000)

[8.3 What to do if my xBRMS is not running 25](#_Toc324501001)

[8.4 Is my xBRMS connecting to its database 26](#_Toc324501002)

[9 Diagnosing IDMS Problems 28](#_Toc324501003)

[9.1 Before You Start 28](#_Toc324501004)

[9.2 Is my IDMS running? 29](#_Toc324501005)

[9.3 What to do if my IDMS is not running 29](#_Toc324501006)

[9.4 Is my IDMS healthy 30](#_Toc324501007)

[9.5 Is my IDMS connecting to its database 31](#_Toc324501008)

[10 Diagnosing JMS Listener Problems 33](#_Toc324501009)

[10.1 Before you start 33](#_Toc324501010)

[10.2 Is my JMS Listener running? 34](#_Toc324501011)

[10.2.1 All counts increase over time 34](#_Toc324501012)

[10.2.2 None of the counts increase over time 35](#_Toc324501013)

[10.2.3 BusinessEvent and/or EntitlementStatus tables’ counts don’t increase over time, but the Event’s table count increases 35](#_Toc324501014)

[10.2.4 Event count doesn’t change but the BusinessEvent and/or EntitlementStatus tables’ counts increase over time 35](#_Toc324501015)

[10.3 What to do if my JMS Listener is not running 35](#_Toc324501016)

[10.4 Is my JMS Listener configured correctly? 36](#_Toc324501017)

# Introduction

## Purpose

This document describes procedures for troubleshooting xBRC problems. The document is intended for a relatively technical audience. The reader is assumed to have at least rudimentary knowledge of Linux in order to be able to perform the more detailed troubleshooting steps.

## Scope

This guide describes procedures for assuring that an xBRC is properly receiving data from functioning xBands (or RFID Cards) and properly forwarding events to upstream applications. It does not address problems with associated systems (for example, GXP or OmniTicket) or with upstream applications that receive xBRC messages (for example, DCL-CDA or Magic Trigger).

## Assumptions and Constraints

N/A

## Definitions

N/A

# xBRC Theory of Operation

In order to troubleshoot problems with the xBRC, it is important to first understand what it does. This section provides a brief description of its operation, focusing on different stages of processing so that subsequent diagnostic techniques have proper context.

## xBRC Architecture Diagram



**Figure 1 - System Design**

## Overall xBRC Data Flow



Figure - xBRC Data Flow

Figure 2 illustrates the flow of data in the xBRC. Reader events (via HTTP) are processed by an embedded HTTP server in the xBRC. These events are added to a doubly-buffered data structure in an asynchronous fashion. The data processing component of the xBRC periodically processes one of the buffers while the other buffer is being filled by the HTTP server. The data processing component logs incoming messages to a low-level file *(/var/log/xbrc/eventdump.txt*) then passes the data to the singulation module. This module also logs its results to the same file. Singulated events are passed to the “modeling” step (differs in different use cases) which ultimate generates messages via the messaging module. Messages are stored in a database and then periodically sent to upstream applications. The messaging module sends messages using the Sonic MQ JMS bus or by using HTTP RESTful protocols.

The embedded HTTP server that is used to process reader events can also process requests from a conventional browser or from xBRMS (refer to the *xBand Reader Monitoring System*). HTTP requests from the xBRC or from the xBRMS can be used to diagnose xBRC problems.

Troubleshooting an xBRC involves verifying data flow in each step and identifying where the process may be misbehaving.

# Diagnosing xBRC Problems

## Before You Start

Before starting any troubleshooting session, collect the following information:

|  |  |
| --- | --- |
| Item | Value |
| IP address of xBRC |  |
| URL of xBRMS |  |
| Login credentials for xBRC |  |

Note that there may be *no* xBRMS or that they may be running on the same machine as the xBRC.

Important: If you don’t have any of items in the table, stop now – there’s nothing you can do. If you have only the first item, you can still perform some diagnostic steps.

The rest of this section is organized into subsections that answer questions about data flow in the xBRC.

## Is my xBRC running?

The easiest way to verify this is by using a browser and pointing it to:

http://<xBRC ip address>:8080/status

If this request times out the xBRC is not running. If it returns an XML document, the xBRC is running.

## What to do if the xBRC is not running

First, verify that the server (physical or virtual) that hosts the xBRC is turned on. You can first try “pinging” the server to see if it’s available over the network. From any Windows or Linux machine on the network type:

ping <xBRC ip address>

If the ping command reports “destination host unreachable”, you will need to consult with an IT resource to restart the xBRC server or otherwise determine why it is not accessible over the network.

If the ping responds successfully, then the xBRC server is running but, for some reason, the xBRC service is not. To start the xBRC service, log into the xBRC server by running *putty* on a Windows machine or *ssh* on a Linux machine. For example, from Linux, if your xBRC is at 10.92.65.5 and your xBRC login user is *testuser* you can use:

ssh testuser@10.92.65.5

Windows *putty* users will enter the xBRC IP address and the login name using interactive screens. Regardless, when prompted for a password, type in the appropriate password for your xBRC login user. At this point, you will be presented with a Linux prompt (the xBRC runs on a Red Hat 5.6 compatible operating system). To verify the status of the xBRC service, type in this command:

$sudo /etc/init.d/xbrc status

Do not type the “$” character – it is the Linux prompt indicating that you are running as a non-privileged user.

If you are prompted for a password, type in your login password again. If the xBRC is not running (as you would expect), the command should return:

No pidfile present

If it *is* running (but somehow, not responding to the /status URL), the command will return:

Service running with pid: ####

If the xBRC is not running, you can start it with:

$sudo /etc/init.d/xbrc start

If it is running, but not responsive, you can restart it with:

$sudo /etc/init.d/xbrc restart

Repeat the “status” command:

$sudo /etc/init.d/xbrc status

to verify that the xBRC is now running.

Having restarted the xBRC, you should now be able to use a browser and browse to the /status URL (as in section 3.1) to verify that it is running.

Note, too, that sometimes it is useful to stop an xBRC, clear out its logs and restart it so that you have “clean” logs to subsequently analyze. You can do this with:

$sudo /etc/init.d/xbrc stop  
$sudo rm –f /var/log/xbrc/\*  
$sudo /etc/init.d/xbrc start

## Is my xBRC “healthy”?

The easiest way to answer this question is to connect a browser to the xBRMS that is monitoring the xBRC:

http://<xBRMS ip address>:8090/XBRMS

Click on the “health” link and find your xBRC in the list presented by the xBRMS. If the xBRMS reports a green light, your xBRC is healthy. If the xBRMS reports a red light it will give you a brief message indicating *why* the xBRC is unhappy. You can also click on the red light to get information about the readers connected to the xBRC. The xBRMS will drill down and show you a list of all the xBR and xTPs associated with the xBRC and will tell you if any of them are not communicating properly.

If you do not have an xBRMS or don’t know how to access it, you can still get some health information by examining the XML output produced by xBRC in response to the /status URL:

http://<xBRC ip address>:8080/status

Examine the value of the <status> tag. If the xBRC is healthy, it will report “green”. If it is not, it will report “red”. If the <status> tag is “red”, look at the value of the <statusMessage> tag. It will tell you why the xBRC is unhappy.

## How to (easily) tell if an xBRC is receiving data from its readers

The best way to answer this question is by logging into the xBRC and looking at the eventdump.txt file. The xBRC logs all low-level and singulation events to this file. By examining this file, you can determine whether the xBRC is receiving events.

As the process of doing this is somewhat complex, here’s a simpler way of examining eventdump.txt. Using a browser, request this URL:

http://<xBRC ip address>:8080/ekgposition

The xBRC refers to eventdump.txt as its “EKG” file (because, if nothing is happening there, the xBRC is “dead”). The xBRC will respond to the URL above by returning a number. This number is the current size of the eventdump.txt file. Refresh your browser. If the xBRC is receiving information (in particular, from xBR readers with a band present), the returned number will be larger. Write down (or copy into the clipboard) the number. Now browse to:

http://<xBRC ip address>:8080/ekg?position=#####&max=100000

Where ###### is the number returned by /ekg?position. The xBRC will return with something similar to:

5424

1324574032289,PROCESS,58

1324574032216,TAP,xpassentry,Bethany Bell,7058b33531610b0d

1324574032217,LRR,entry-4,Jill Neusch,38bb1a5a5dd5b83d,0,-52,2476,1

1324574032217,LRR,entry-4,Jill Neusch,38bb1a5a5dd5b83d,0,-52,2476,0

1324574032217,LRR,entry-4,Cedric Douglas,2a12e9ef5d4b252a,0,-45,2401,0

1324574032217,LRR,entry-4,Cedric Douglas,2a12e9ef5d4b252a,0,-45,2401,1

1324574032217,LRR,entry-4,Theodore Dauila,3d85302178ba59d6,0,-56,2476,0

1324574032217,LRR,entry-4,Theodore Dauila,3d85302178ba59d6,0,-56,2476,1

1324574032217,LRR,entry-3,Cedric Douglas,2a12e9ef5d4b252a,0,-55,2401,0

1324574032217,LRR,entry-3,Cedric Douglas,2a12e9ef5d4b252a,0,-55,2401,1

1324574032217,SNG,Entry,2a12e9ef5d4b252a,Only

1324574032217,SNG,Entry,38bb1a5a5dd5b83d,Only

1324574032217,SNG,Entry,3d85302178ba59d6,Only

If the xBRC does *not* return lines with “TAP” or “LRR” in them, it is not receiving reader events.

The first number is the new file size of eventdump.txt. If you want to continue tracking eventdump.txt, you can issue the same URL but using this number as the new position= argument.

The second and subsequent lines describe xBRC event and singulation activity. The format of each line is:

<time stamp>,<record type>,<payload>

where <time stamp> is in the form of “epoch millisecond time” (number of milliseconds after January 1st, 1970 00:00:00 UTC. <payload> varies according to the <record type>. The different record types are:

**PROCESS** – indicates how many reader events are being processed by the xBRC in the current batch.

**TAP** – indicates an xTP tap event. The payload describes the name of the reader that reported the event, the name of the guest associated with the tapped band, and the RFID (tap id) of the band.

**LRR** – indicates an xBR long range read event. The payload describes the name of the reader that reported the event, the name of the guest associated with the transmitting band, the long-range id of the band, the transmission’s “packet number” (cycles from 0-255 repeatedly), the signal strength in dB (typically -90 to -40), the frequency and the radio “bank” (0 or 1) that picked up the transmission in the xBR.

**SNG** – indicates an xBRC singulation. The payload identifies the reader location where the guest has been determined to be, the long-range id of the (one of the) guest’s band(s) and the “basis” for the singulation. “Only” indicates that only one reader location was reporting data for the guest. Other bases included “MeanSS”, “PeakSS”, “Count” and “First”.

## Another way to tell if an xBRC is receiving data from its readers

If you are Linux savvy, you have more tools available for examining the eventdump.txt file if you log into the xBRC and use command line tools.

Using putty or ssh, log into the xBRC as described in 3.2. Now, type this command to monitor eventdump.txt:

$sudo tail –f /var/log/xbrc/eventdump.txt

Again, don’t type the “$” and *do* type in your login password if prompted to by *sudo*.

The *tail* command will display all data being added to eventdump.txt until you terminate the command with Ctrl-C. If you tap a band to an xTP or if you have bands that are transmitting close to an xBR, you will “TAP” and “LRR” records appearing as described in the previous section. If you don’t see TAP or LRR records, the xBRC is not receiving any data from its readers.

Combining the *tail* command with the *grep* command can be useful. For example:

$sudo tail –f /var/log/xbrc/eventdump.txt | grep TAP

This command will show you only tap events. This one:

$sudo tail –f /var/log/xbrc/eventdump.txt | grep LRR

will show you only long-range reads. If you’re interested in long range reads for a particular band, you can grep for its long-range id, for example:

$sudo tail –f /var/log/xbrc/eventdump.txt | grep 2a12e9ef5d4b252a

The *grep* command supports sophisticated regular expressions that can be used for a variety of purposes. Please refer to external grep documentation if you are interested in the topic.

## Are Band IDs being properly mapped to guests?

The entries in eventdump.txt include references to guest names and to long-range and RFIDs. If the xBRC is properly interacting with IDMS, these entries will appear as described in section 3.4. LRR and TAP entries will display the name of the guest associated with the band involved in the entry. If the *wrong* name is displayed, the IDMS contains incorrect information in its database and needs to be corrected.

You can tell if the xBRC is encountering IDMS id mapping issues by looking at the entries in the eventdump.txt file. If you see a guest name that looks like:

?BAND=########

This indicates that the IDMS mapped a long-range or RFID to a band but that the band is not associated with any guest. The xBRC will log the low-level event, but will then ignore it in any subsequent processing.

If, however, you see a guest name that looks like:

?LRID=########

or:

?RFID=########

this indicates that the IDMS does not recognize an LRID (long-range) or RFID (tap) as valid for any band in its database. Since it can’t associate the id with a band, it also can’t associate the band with a guest.

## Is my xBRC is singulating guests?

One way to do this is to follow the instructions in section 3.5 and use this command:

$sudo tail –f /var/log/xbrc/eventdump.txt | grep SNG

This will show you the singulation output in the eventdump.txt file.

Another way, however, is to look at the Attraction View (called “Facilities View” in more recent versions of the xBRC). Using a browser, visit this URL:

http://<xBRC ip address>:8090/UI

Notice the port number is 8090 not 8080 as used in other URLs. The xBRC UI’s home page presents several links one of which will be the “Attraction View” or “Facilities View”. Click on this link to see the “subway diagram” associated with your xBRC. If the xBRC is being used for an attraction, the subway diagram will contain various “stations”. If it’s being used for DCL or restaurants or other uses, it may have a very simple subway diagram. Regardless, the subway diagram will tell you what guests the xBRC considers to be present in the facility. Numbers and icons will appear indicating where/how many guests have been sensed. Exactly how this appears depends on the particular xBRC deployment. The subway diagram is customized to particular facilities (by using the “Facility Designer” part of the xBRC UI).

Note that you can click on the guest icons or on the numbers that appear in the subway diagram to identify which guests are at which locations. When the guest list appears, if you click on the magnifying glass next to a particular guest, the xBRC UI will present the “power level” display that tells you what long-range read signal strengths are being read for the chosen guest as well as the singulation output for that guest.

## Is my xBRC generating messages?

Before the xBRC sends a message upstream (via SonicMQ or via HTTP), it first stores that message in its local database. You can tell if an xBRC is generating messages by examining the output of:

http://<xBRC ip address>:8080/status

Note the value reported by the <messageCount> tag. As guests tap bands and move throughout your facility, the xBRC should generate messages and the <messageCount> variable should increase.

## How to view xBRC messages

The messages generated by the xBRC and stored in its database can be viewed from a browser by visiting:

http://<xBRC ip address>:8080/messages

This URL will return ALL of the messages stored by the xBRC, including old ones that might have previously already been sent upstream. This may be a lot of data. You can restrict the amount of data returned by including additional parameters:

http://<xBRC ip address>:8080/messages?after=#####&max=###

The *after* parameter tells the xBRC to return only messages whose “sequence number” is greater than the provided number. The *max* argument restricts the number of messages returned.

Note that the output of the /status URL (see the previous section) also includes a tag named <lastMessageSeq>. This tag identifies the sequence number of the last message written to the xBRC database. You can use this value when composing a /messages URL in order to obtain recent messages (e.g. the previous 100 messages).

## How to tell whether an xBRC is configured to send messages to Sonic MQ or to an HTTP destination

You can tell if an xBRC is configured to send upstream messages via HTTP by looking at the output of the /status URL:

http://<xBRC ip address>:8080/status

If the <updateStreamURL> tag contains a valid URL (not starting with a # sign), then the xBRC will use HTTP RESTful protocols to send messages to that URL.

Alas, the /status output does not currently contain information about Sonic MQ (this will be corrected soon). To determine if an xBRC is configured to send messages to SonicMQ use a browser to visit this URL:

http://<xBRC ip address>:8080/currentconfiguration

This will return large output. Ignore everything except the tags that look like this:

<property class="ESBInfo" name="jmsbroker">#</property>

If the value of this property is set to an ip address (and port), then the xBRC is configured to send messages to Sonic MQ. Alternatively, if the value is not set or if it is set to “#”, as in this example, then the xBRC is not configured to send messages to SonicMQ.

## Are messages being sent upstream?

Again, the output of the /status URL provides important clues. Two tags are included, <lastMessageToJMS> and <lastMessageToUpdateStream>. If the xBRC is configured to write messages to Sonic MQ, the value reported by the <lastMessageToJMS> tag should regularly increase. Alternatively, if the xBRC is configured to send messages via HTTP, the value reported by <lastMessageToUpdateStream> should increase.

As described in section 3.9, the <lastMessageSeq> tag reported by the /status URL identifies the most recent message sequence number written to the xBRC database. If this number is the same as that reported by <lastMessageToJMS> or <lastMessageToUpdateStream> then the xBRC is properly sending messages upstream.

If the number reported by the <lastMessageToJMS> tag is not increasing, but your xBRC is configured to send messages to Sonic MQ instead of via HTTP, make sure that the value of the LastMessageIdToJMS field in the xBRC’s Status database table equals the value in the <lastMessageSeq> tag. When the LastMessageIdToIMS is smaller than <lastMessageSeq>, the messages do not get sent. To verify:

Log in to xBRC’s MySQL database (enter password when prompted):

# mysql -u EMUser -p

Switch to using Mayhem database:

mysql> use Mayhem;

Check what the value of LastMessageIdToJMS is:

mysql> select \* from Status;

+---------------------------+-------------------------+

| Property | Value |

+---------------------------+-------------------------+

| LastStateStore | 2012-05-11T18:25:16.323 |

| LastMessageIdToJMS | 61 |

| LastMessageIdToPostStream | -1 |

+---------------------------+-------------------------+

Set the value of LastMessageIdToJMS to -1 if you find it to be smaller than <lastMessageSeq>. This will cause the xBRC to resend all the JMS messages.

mysql> update Status set Value = -1 where Property = “LastMessageIdToJMS”;

Restart the xBRC:

# /etc/init.d/xbrc restart

## How can I tell if an xBRC is reporting errors?

The xBRC writes log files to its /var/log/xbrc directory (the same place where the eventdump.txt file is written.) Most important is the /var/log/xbrc/xbrcController.log file. You can view this file by logging into the logging into the xBRC and using the following command:

$sudo less /var/log/xbrc/xbrcController.log

Note, however, that this file is a *rotating* log file and, when it gets large, it gets split into a separate file named /var/log/xbrc/xbrcController.log.1. Older files are maintained in .2, .3, etc. If you are looking for a particular error message, the *grep* command is useful for searching through multiple error files.

By default, only warnings and errors are written to the xbrcController.log file. Sometimes, it is useful to increase the amount of information written to the file in order to better diagnose an anomalous condition. The amount of information written to the log file is controlled by the /usr/share/xbrc/log4j.xml file. A fragment of this file reads:

<logger name=”com.disney.xband” additivity=”false”>  
 <level value=”warn”/>  
 <appender-ref ref=”xbrc” />  
</logger>

If you change “warn” to “debug”, the xBRC will generate significantly more output in the xbrcController.log file(s). Note that after changing this file (using some Linux text editor), you will need to restart the xBRC in order for the change to take effect:

$sudo vi /usr/share/xbrc/log4j.xml  
$sudo /etc/init.d/xbrc restart

# Diagnosing xBR Problems

## Before You Start

Before starting any troubleshooting session, collect the following information:

|  |  |
| --- | --- |
| Item | Value |
| IP address of xBR – labeled on case exterior |  |

## The xBRC is OK, but I still have problems

You have determined by going through the xBRC troubleshooting that the xBRC is OK and that the problem seems to be one or multiple xBRs. There are several things that could be wrong with the xBR. Not in any particular order of potential.

1. The xBR Ethernet could be faulty
2. The xBR application could be stopped
3. The receive radio/s could be faulty
4. The transmit radio could be faulty (not possible currently since this radio is not being used)

## Faulty xBR Ethernet

The ability to communicate with the xBR should be established first. The xBRC troubleshooting above will determine if the xBR is “happy” by examining the xBRMS health link and looking for red lights on any of the xBR’s. The xBR transmit a health packet every second to the xBRC so if this does not happen the xBR is flagged red. The Ethernet link could still be fine.

To access whether the problem is an Ethernet problem or not, the xBR should be pinged. If the ping is returned regularly, the Ethernet is ok. If not, the cable should be checked before removing the xBR.

## The LRR Application is stopped

Open a Putty window emulating an ssh session with the IP address of the suspect xBR. The login is “root” and no password.

Type the Linux command “ps”. You should get a list that looks something like this.

PID USER VSZ STAT COMMAND

1 root 1708 S init [5]

2 root 0 SW [kthreadd]

3 root 0 SW [ksoftirqd/0]

4 root 0 SW [kworker/0:0]

5 root 0 SW [kworker/u:0]

6 root 0 SW [rcu\_kthread]

7 root 0 SW< [khelper]

8 root 0 SW [irq/72-serial i]

9 root 0 SW [irq/73-serial i]

10 root 0 SW [irq/74-serial i]

11 root 0 SW [sync\_supers]

12 root 0 SW [bdi-default]

13 root 0 SW< [kblockd]

14 root 0 SW< [omap2\_mcspi]

15 root 0 SW [khubd]

16 root 0 SW< [twl4030-irqchip]

17 root 0 SW [twl4030-irq]

18 root 0 SW [irq/378-twl4030]

19 root 0 SW [kworker/u:1]

20 root 0 SW< [rpciod]

21 root 0 SW [khungtaskd]

23 root 0 SW [kswapd0]

24 root 0 SW [fsnotify\_mark]

25 root 0 SW< [nfsiod]

26 root 0 SW< [crypto]

34 root 0 SW [irq/371-twl4030]

37 root 0 SW [mtdblock0]

38 root 0 SW [mtdblock1]

39 root 0 SW [mtdblock2]

40 root 0 SW [mtdblock3]

41 root 0 SW [mtdblock4]

46 root 0 SW [irq/379-rtc0]

48 root 0 DW [mmcqd/0]

50 root 0 SW [kjournald]

72 root 2112 S < /sbin/udevd -d

157 root 0 SW [flush-179:0]

1220 root 2336 S dhclient -pf /var/run/dhclient.eth0.pid eth0

1264 root 192m S /usr/bin/grover

1343 root 4432 S /usr/sbin/sshd

1357 root 3004 S /sbin/syslogd -n -C64 -m 20

1364 root 2940 S /sbin/klogd -n

1368 root 2672 S login -- root

1369 root 1968 S /sbin/getty 38400 tty1

1432 root 1704 S /usr/bin/netplugd -p /var/run/netplugd.pid

1498 root 3288 S -sh

1559 root 0 SW [kworker/0:1]

1576 root 0 SW [kworker/0:2]

1578 root 4488 S sshd: root@pts/0

1582 root 3288 S -sh

1584 root 3288 R ps

The LRR Application is running if you see the line */usr/bin/grover* in the list. If this is not there the LRR Application has either shut down or has not restarted after a power interruption.

# xTP New Unit Testing

An xTPE unit may be tested for functionality in the following way.

1. Plug in the DC power and Ethernet cables to the xTPE unit.
2. The xTPE takes approximately a minute to boot successfully. The xTPE is booted and ready to communicate when the Ethernet link/activity LEDs are blinking.
3. On your computer, open the PuTTY application or other communications software to establish a secure shell connection to the xFPE.
   1. If using PuTTY, select **SSH** for the Connection Type
   2. Use the default IP address. If no DHCP server is running, i.e. if you're connecting the unit directly to your computer) is 169.254.0.2
4. Hit "Open"
5. (Saving this IP address as a saved session in PuTTY can save a lot of time. I highly recommend it. You can do this by typing in the IP address and hitting "Save" next to the "Saved Sessions" dialog.
6. A terminal window should appear; if your computer raises any security concerns here, hit Ok/Yes/Continue. The xTPE should not hack your system and corrupt everything you own.
7. For the login name, type root

You're connected!

To test a new unit, do the following:

1. Power up

2. Verify Ethernet link/activity LEDS are blinking after the ~1 minute boot process. (assuming the unit is plugged into an Ethernet pair (either a switch or directly to a host)

3. Assuming there is no DHCP server running, SSH into 169.254.0.2, user is root, no password.

4. Set MAC address (only needs to be done to new units)

5. Stop the dap-reader process

6. Run the LED tests

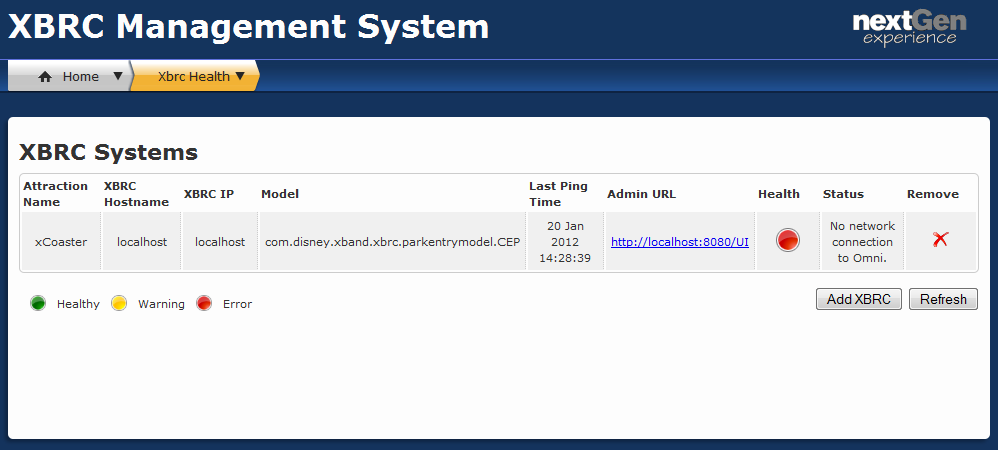
7. Run the sensortest command with lights on and off

8. Run the speaker test

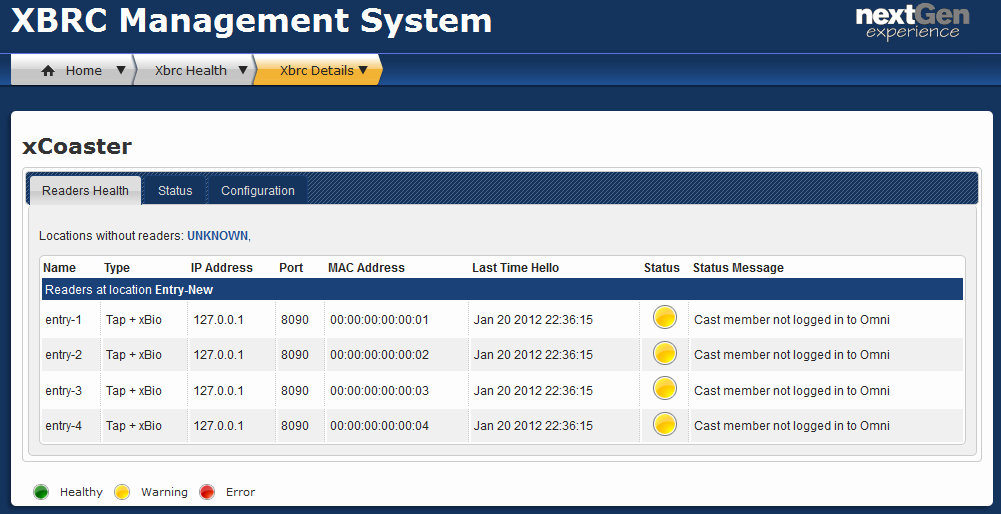
9. Run the dap-reader application and verify it reads an xBand or RFID card.

# xBRMS Health Page

The xBRMS shows health status for one or more configured xBRCs and their readers. The health status consists of a status message plus the severity of the message. The severity may be Normal, represented by a green icon, Warning, represented by a yellow icon, or Error, represented by a red icon. In the example screen capture below, there is an Error message showing “No network connection to Omni”.



In addition to the overall xBRC health status, the xBRMS can display health status for each reader. The reader health status page is accessible by clicking on the circular health icon, the red circle in the screen shot above. In the example screen capture below all readers report a warning message “Cast member not logged in to Omni”.



## xBRMS Health Status Messages

The following table lists health status messages that may appear on the xBRMS health status pages for the xBRC or for each reader. Next to each message there is a short description of its meaning and a possible solution to the problem. These messages are common to all the models.

| Status Message | Sev. | Description | Causes/Resolutions |
| --- | --- | --- | --- |
| Reader <reader id> not communicating | Error | xBRC did not receive a “hello” message on time from this reader. | Causes: 1) reader turned off, 2) reader has no network access, 3) reader physically disconnected. Solution: ping reader IP or ssh to reader. |
| xBRC version 1.0.0.0 is smaller than required by reader 1.0.0.1 | Warn | Readers require that they are connecting to an xBRC having some minimum version number. | Cause: Most likely reader software was upgraded, but the xBRC software is still running an older version. Solution: upgrade the xBRC to the latest version. |
| Reader version 1.0.0.0 is smaller than required by xBRC | Warn | xBRC requires that each reader has some minimum version number. No reader events from this reader are processed until the reader software is upgraded. | Cause: The xBRC maintains a local repository of compatible software packages for the readers. If a reader software version is too low, the xBRC will automatically attempt to upgrade the reader software. If the upgrade operation is successful then the status message should disappear shortly. If the status message persists, there most likely was an error upgrading the reader software. Solution: upgrade the reader software to the latest version. |

# Diagnosing Park Entry Problems

Before attempting to diagnose any Park Entry Problems, please refer to sections [XBRMS Health Page](#_XBRMS_Health_Page) and [Diagnosing xBRC Problems](#_Diagnosing_xBRC_Problems) to make sure that the xBRC is operational and processing reader events.

## Park Entry XBRMS Health Status Messages

The following table lists health status messages that may appear on the XBRMS health status pages either for the xBRC or for each reader. Next to each message there is a short description of its meaning and a possible solution to the problem.

| Status Message | Sev. | Description | Causes/Resolutions |
| --- | --- | --- | --- |
| No network connection to OmniTicket. | Error | The xBRC could not establish TCP/IP connection to the OmniTicket system. The IP address and port are configurable in the xBRC Config table: omniticketadderss and omniticketport. | Causes: 1) no network access, 2) OmniTicket system down, 3) OmniTicket system has no network access, 4) xBRC configuration parameters omniticketaddress and omniticketport are not correctly set. Solution: 1) check xBRC access to the network (ping Omni server IP), 2) check if Omni system is running, 3) make sure OmniTicket server can connect to the network, 4) verify that the omniticketaddress and omniticketport are correct. |
| No connection to the Cast App socket server. | Error | The xBRC communicates with the Trifecta Cast App socket server over TCP/IP. The IP address and port are configurable in the xBRC Config table: castappserveraddress and castappserverport. | Causes: 1) cast app server is not running, 2) the xBRC configuration parameters castappserveraddress and castappserverport are not correctly set. Solution: 1) start the cast app server, 2) verify that castappserveraddress and castappserverport are correct. |
| No connection to the Cast App observer socket server. | Error | The xBRC communicates with the Trifecta Cast App observer socket server over TCP/IP. The IP address and port are configurable in the xBRC Config table: castappserverobserveraddress and castappserverobserverport. | Causes: 1) cast app observer server is not running, 2) the xBRC configuration parameters castappserverobserveraddress and castappserverobserverport are not correctly set. Solution: 1) start the cast app server, 2) verify that castappserverobserveraddress and castappserverobserverport are correct. |
| Cast member not logged into OmniTicket system at location Entry. | Warn | A cast member must log into a reader location before the xBRC can process events from readers at that location. | Cause: no cast member logged into the reader location. Solution: log in the cast member. |
| Waiting for login Status response from OmniTicket. | Warn | The status message is sent to the OmniTicket system by the xBRC on behalf of each reader to find out if a cast member is already logged into a reader. | Cause: OmniTicket did not respond to the status message. Solution: No user action is required. This warning condition will either time out or will be cleared when OmniTicket response is received. |
| Waiting for Login/Logoff response from OmniTicket. | Warn | A login or logoff message is sent to OmniTicket when a cast member logs in/off to/from a location. | Cause: OmniTicket did not respond to the login/logoff message. Solution: No user action is required. The error will either time out or will be cleared when OmniTicket response is received. |
| Waiting for Connect response from OmniTicket. | Warn | A connect message is sent to the OmniTicket system to connect a reader. | Cause: OmniTicket did not respond to the connect message. Solution: No user action is required. The error will either time out or will be cleared when OmniTicket response is received. |
| OmniTicket rejected user logon for user <user name> using RFID <rfid>. | Error | This error is self- explanatory. | Solution: The CastMember table contains the omniUsername and omniPassword for each cast member that logs into OmniTicket. Make sure that the omniUsername and omniPassword are correct and that the OmniTicket system is configured to accept this user. |
| Trying to connect to OmniTicket | Warn | Reader status | Solution: see the xBRC status message above “Waiting for Connect response from Omni.” |
| Waiting for login status response from OmniTicket | Warn | Reader status | Solution: see the xBRC status message above “Waiting for login Status response from Omni.” |
| Cast member not logged in to OmniTicket. | Warn | Reader status | Solution: see the xBRC status message above “Cast member not logged into Omni system at location Entry.” |
| Awaiting Cast member login response from OmniTicket. | Warn | Reader status | Solution: see the xBRC status message above “Waiting for Login/Logoff response from Omni.” |

# Diagnosing xBRMS Problems

## Before You Start

Before starting any troubleshooting session, collect the following information:

|  |  |
| --- | --- |
| Item | Value |
| IP address of xBRMS (or DNS name) | XBRMS-SERVER |
| xBRMS HTTP port | XBRMS-PORT |
| xBRMS login user | XBRMS-USER |
| xBRMS login password | XBRMS-PASSWORD |
| IP address of xBRMS database | XBRMS-DB-SERVER |
| xBRMS database user | XBRMS-DB-USER |
| xBRMS database password | XBRMS-DB-PASSWORD |

Throughout the rest of the xBRMS troubleshooting section values from the above table are used as place holders for the actual values of the xBRMS you are troubleshooting.

Important: If you don’t have any of items in the table, stop now – there’s nothing you can do. If you have only the first item, you can still perform some diagnostic steps.

## Is my xBRMS running?

To check if your xBRMS is running, access this URL:

<http://XBRMS-SERVER:XBRMS-PORT/XBRMS>

If your xBRMS is running, you’ll see the home page:

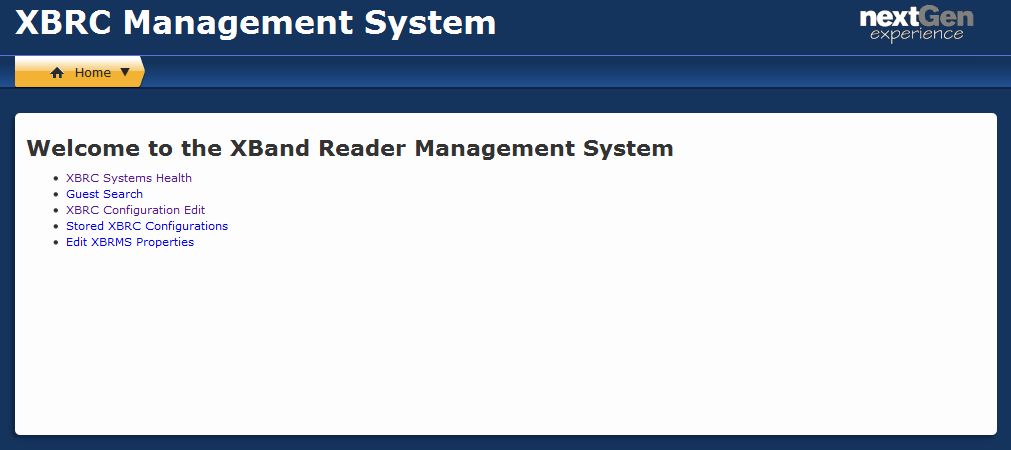


Figure : xBRMS Home Page

If you get a “server not found” message, your xBRMS is not running. Follow instructions in the [What to do if my xBRMS is not running](#_What_to_do) section of this guide to diagnose and fix this issue.

If you see the setup screen, your xBRMS is running, but it is not able to contact its database. To fix this problem, follow instructions in the [Is my xBRMS connecting to its database](#_Is_my_xBRMS) section of this guide.

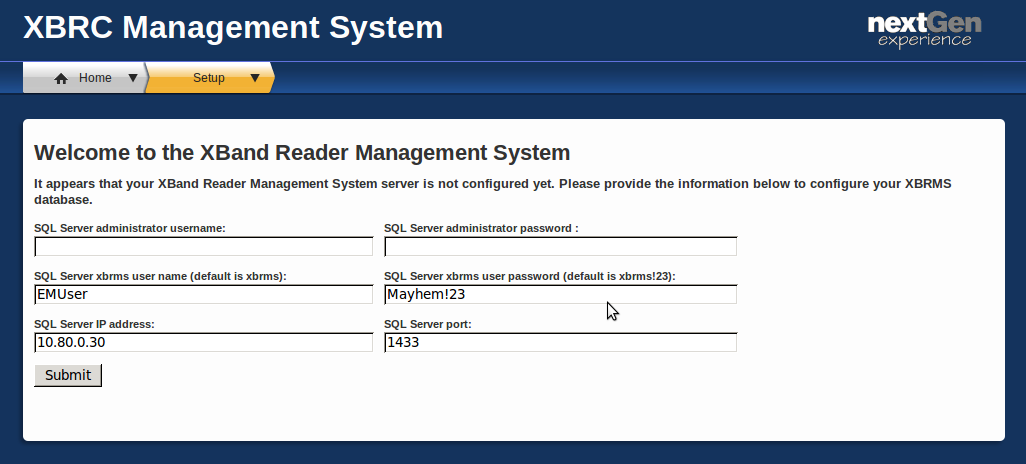


Figure : xBRMS Setup Screen

## What to do if my xBRMS is not running

Verify that the physical/virtual server that hosts the xBRMS is running:

ping XBRMS-SERVER

If the ping command reports “destination host unreachable”, you will need to consult with an IT resource to restart the xBRMS server or otherwise determine why it is not accessible over the network.

If the ping responds successfully, then the xBRMS server is running but, for some reason, the xBRMS service is not. To start the xBRMS service, log into the xBRMS server by running *putty* on a Windows machine or *ssh* on a Linux machine. For example, from Linux, execute this command:

ssh XBRMS-USER@XBRMS-SERVER

Windows *putty* users will enter the XBRMS-SERVER address and the XBRMS-LOGIN name using interactive screens. Regardless, when prompted for a password, type in the appropriate XBRMS-PASSWORD for your xBRMS login user. At this point, you will be presented with a Linux prompt (the xBRMS runs on a Red Hat 5.6 compatible operating system). To verify the status of the xBRMS service, type in this command (you have to be superuser to run it):

#sudo /etc/init.d/bootssptcserver1 status

Do not type the “#” character – it is the Linux prompt indicating that you are running as a superuser.

If you are prompted for a password, type in your XBRMS-PASSWORD again. If the xBRMS is not running (as you would expect), the command should return:

No pidfile present

If it *is* running (but somehow, you are getting “server not found” messages when trying to access http:// XBRMS-SERVER:XBRMS-PORT/XBRMS), the command will return:

Service running with pid: ####

If the xBRMC is not running, you can start it with:

#sudo /etc/init.d/bootssptcserver1 start

If it is running, but not responsive, you can restart it with:

#sudo /etc/init.d/bootssptcserver1 restart

Repeat the “status” command:

#sudo /etc/init.d/bootssptcserver1 status

to verify that the xBRMS is now running. Having restarted the xBRMS, you should now be able to use a browser and browse to the http:// XBRMS-SERVER:XBRMS-PORT/XBRMS URL to verify that it is running.

Note, too, that sometimes it is useful to stop an xBRMS clear out its logs and restart it so that you have “clean” logs to subsequently analyze. You can do this with:

#sudo /etc/init.d/bootssptcserver1 stop  
#sudo rm –f /opt/apps/tcserverApp/tcServer-6.0/ssp-tcserver1/logs/\*  
#sudo /etc/init.d/bootssptcserver1 start

## Is my xBRMS connecting to its database

To check if your xBRMS is connecting to its database, access this URL:

<http://XBRMS-SERVER:XBRMS-PORT/XBRMS>

If your xBRMS is connecting, you’ll see the home page. Otherwise, you’ll see the setup page:

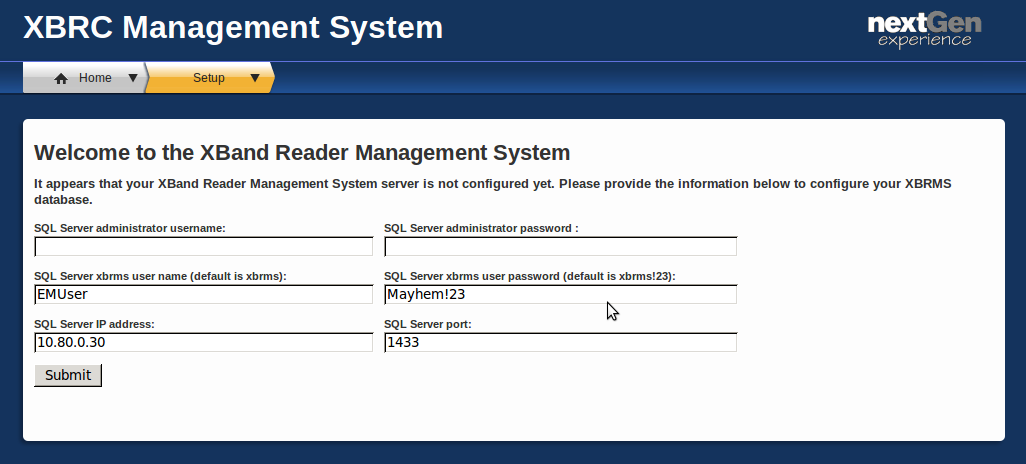


Figure : xBRMS Setup Page

Type in correct connection information into the fields provided and click submit.

xBRMS will attempt to connect to its database with using the new connection information. If the information you provided is correct, you will see this message: “Connection to the server was successful. You may now proceed to the home page.”

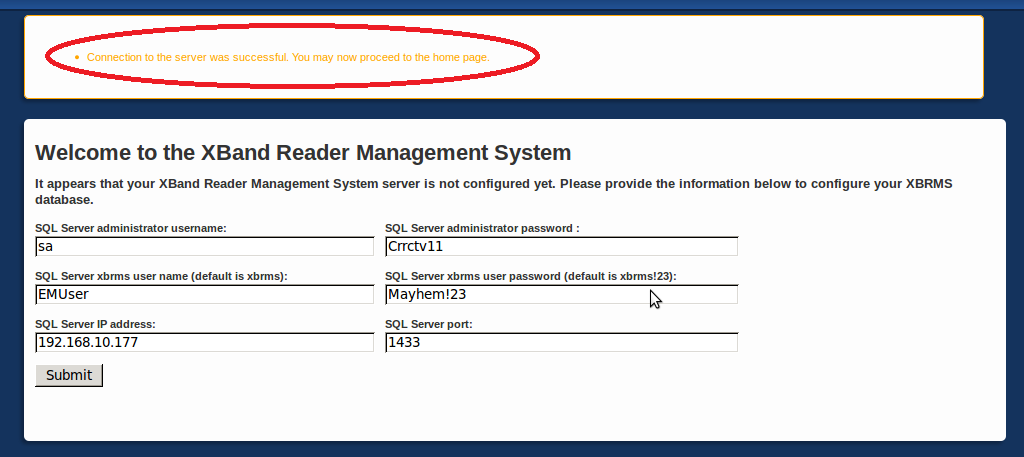


Figure : xBRMS Setup Page Success Message

At this point, you should be able to access the home page:  
<http://XBRMS-SERVER:XBRMS-PORT/XBRMS>

If you provide incorrect database connection information, xBRMS will still not be able to connect to its database, in which case you will get a “There was an error applying your changes. Network error IOException: connection refused” message:

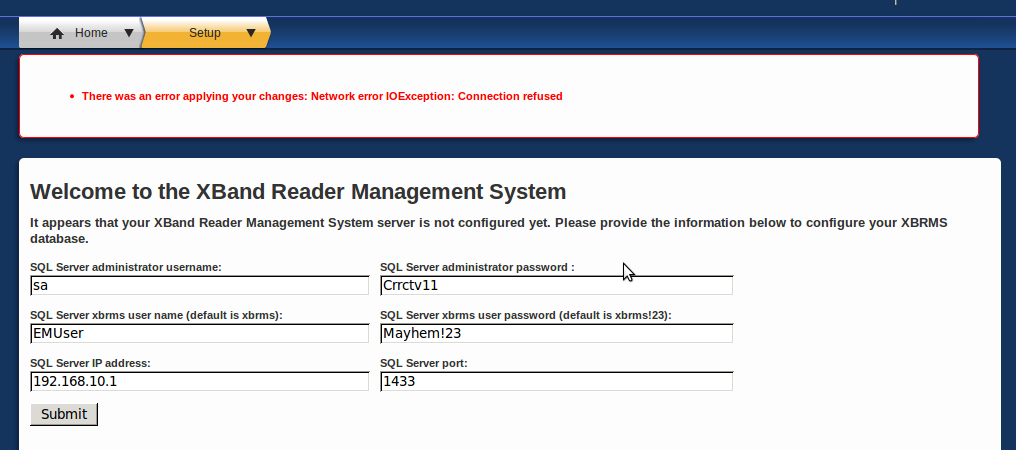


Figure : xBRMS Setup Page Failure Message

# Diagnosing IDMS Problems

## Before You Start

Before starting any troubleshooting session, collect the following information:

|  |  |
| --- | --- |
| Item | Value |
| IDMS IP address (or DNS name) | IDMS-SERVER |
| IDMS HTTP port | IDMS-PORT |
| IDMS login user | IDMS-USER |
| IDMS login password | IDMS-PASSWORD |
| IDMS database name | IDMS-DB-NAME |
| IDMS database IP address | IDMS-DB-SERVER |
| IDMS database user | IDMS-DB-USER |
| IDMS database password | IDMS-DB-PASSWORD |

Throughout the rest of the IMDS troubleshooting section values from the above table are used as place holders for the actual values of the IDMS you are troubleshooting.

Important: If you don’t have any of items in the table, stop now – there’s nothing you can do. If you have only the first item, you can still perform some diagnostic steps.

## Is my IDMS running?

The easiest way to verify this is by using a browser and pointing it to:

http://IDMS-SERVER:IDMS-PORT/IDMS/status

If this request times out the IDMS is not running. If it returns an XML document, the IDMS is running.

If the XML returned shows status as Green, your IDMS is healthy.

<status>

<hostname>nl-flfa-00073</hostname>

<startTime>2012-04-27T13:31:34.177</startTime>

<status>**Green**</status>

<statusMessage/>

<version>1.0.0.1527</version>

</status>

If the XML returned shows status as Red, your IDMS is not able to connect to its database.

<status>

<hostname>nl-flfa-00073</hostname>

<startTime>2012-04-27T13:31:34.177</startTime>

<status>**Red**</status>

<statusMessage>**No database connection: No connection**</statusMessage>

<version>1.0.0.1527</version>

</status>

## What to do if my IDMS is not running

Verify that the physical/virtual server that hosts the IDMS is running:

ping IDMS-SERVER

If the ping command reports “destination host unreachable”, you will need to consult with an IT resource to restart the IDMS server or otherwise determine why it is not accessible over the network.

If the ping responds successfully, then the IDMS server is running but, for some reason, the IDMS service is not. To start the IDMS service, log into the IDMS server by running *putty* on a Windows machine or *ssh* on a Linux machine. For example, from Linux, execute this command:

ssh IDMS-USER@IDMS-SERVER

Windows *putty* users will enter the IDMS-SERVER address and the IDMS-LOGIN name using interactive screens. Regardless, when prompted for a password, type in the appropriate IDMS-PASSWORD for your IDMS login user. At this point, you will be presented with a Linux prompt (the IDMS runs on a Red Hat 5.6 compatible operating system). To verify the status of the IDMS service, type in this command (you have to be superuser to run it):

#sudo /etc/init.d/bootssptcserver1 status

Do not type the “#” character – it is the Linux prompt indicating that you are running as a superuser.

If you are prompted for a password, type in your IDMS-PASSWORD again. If the xBRC is not running (as you would expect), the command should return:

No pidfile present

If it *is* running (but somehow, you are getting “server not found” messages when trying to access http://IDMS-SERVER:IDMS-PORT/IDMS), the command will return:

Service running with pid: ####

If the IDMS is not running, you can start it with:

#sudo /etc/init.d/bootssptcserver1 start

If it is running, but not responsive, you can restart it with:

#sudo /etc/init.d/bootssptcserver1 restart

Repeat the “status” command:

#sudo /etc/init.d/bootssptcserver1 status

to verify that the IDMS is now running. Having restarted the IDMS, you should now be able to use a browser and browse to the http://IDMS-SERVER:IDMS-PORT/IDMS/status URL to verify that it is running.

Note, too, that sometimes it is useful to stop an IDMS clear out its logs and restart it so that you have “clean” logs to subsequently analyze. You can do this with:

#sudo /etc/init.d/bootssptcserver1 stop  
#sudo rm –f /opt/apps/tcserverApp/tcServer-6.0/ssp-tcserver1/logs/\*  
#sudo /etc/init.d/bootssptcserver1 start

## Is my IDMS healthy

The easiest way to answer this question is to connect a browser to the xBRMS that is monitoring your IDMS:

http://XBRMS-SERVER:XBRMS-PORT/XBRMS

Click on the “health” link and find your IDMS in the list presented by the xBRMS. If the xBRMS reports a green light, your IDMS is healthy. If the xBRMS reports a red light it will give you a brief message indicating *why* the IDMS is unhappy. You can also click on the red light to get more information about the IDMS’s status and to configure IDMS.



Figure : IDMS Status Page in xBRMS



Figure : IDMS Configuration Page in xBRMS

If you do not have an xBRMS or don’t know how to access it, read the [Is my xBRMS running](#_Is_my_xBRMS_1). It will tell you how to access it.

## Is my IDMS connecting to its database

The easiest way to verify this is by using a browser and pointing it to:

http://IDMS-SERVER:IDMS-PORT/IDMS/status

If this request times out the IDMS is not running and you should follow steps in [What to do if my IDMS is not running](#_What_to_do_1) section.

If the call returns an XML document and the XML returned shows status as Green, your IDMS is connecting to its database.

<status>

<hostname>nl-flfa-00073</hostname>

<startTime>2012-04-27T13:31:34.177</startTime>

<status>**Green**</status>

<statusMessage/>

<version>1.0.0.1527</version>

</status>

If the XML returned shows status as Red, your IDMS is not able to connect to its database.

<status>

<hostname>nl-flfa-00073</hostname>

<startTime>2012-04-27T13:31:34.177</startTime>

<status>**Red**</status>

<statusMessage>**No database connection: No connection**</statusMessage>

<version>1.0.0.1527</version>

</status>

If the XML returned shows status as Red, your IDMS is not able to connect to its database. To fix this problem, connect a browser to the xBRMS that is monitoring your IDMS:

http://XBRMS-SERVER:XBRMS-PORT/XBRMS

Click on the “health” link and find your IDMS in the list presented by the xBRMS. Click on the red or green light, and then switch to the Configuration tab.



Figure : IDMS Configuration Page in xBRMS

Provide configuration values listed below and click Submit:

Database URL: *jdbc:jtds:sqlserver://IDMS-DB-SERVER:IDMS-DB-PORT/IDMS-DB-NAME*

Database User Name: *IDMS-DB-USER*

Database Password: *IDMS-DB-PASSWORD*

If you do not have an xBRMS or don’t know how to access it, read the [Is my xBRMS running](#_Is_my_xBRMS_1). It will tell you how to access it.

# Diagnosing JMS Listener Problems

## Before you start

Before starting any troubleshooting session, collect the following information:

| Item | Value |
| --- | --- |
| JMS Listener IP address (or DNS name) | JMSLIST-SERVER |
| JMS Listener HTTP port | JMSLIST-PORT |
| JMS Listener login user | JMSLIST-USER |
| JMS Listener login password | JMSLIST-PASSWORD |
| xBRMS database server IP or DNS name | XBRMS-DB-IP |
| xBRMS database port | XBRMS-DB-PORT |
| xBRMS database name | XBRMS-DB-NAME |
| xBRMS database user | XBRMS-DB-USER |
| xBRMS database password | XBRMS-DB-PASSWORD |
| IDMS IP or DNS name | IDMS-SERVER |
| IDMS Port | IDMS-PORT |
| GxP JMS Broker IP | GXP-BROKER-SERVER |
| GxP JMS Broker port | GXP-BROKER-PORT |
| GxP JMS Broker topic | GXP-BROKER-TOPIC |
| GxP UMS user name | GXP-BROKER-USER |
| GxP JMS Broker password | GXP-BROKER-PASSWORD |
| xBRC JMS Broker IP | XBRC-BROKER-SERVER |
| xBRC JMS Broker port | XBRC-BROKER-PORT |
| xBRC JMS Broker topic | XBRC-BROKER-TOPIC |
| xBRC JMS user name | XBRC-BROKER-USER |
| xBRC JMS Broker password | XBRC-BROKER-PASSWORD |
| GXP IP address or DNS name | GXP-SERVER |
| GXP port | GXP-PORT |

Throughout the rest of the JMS Listener troubleshooting section values from the above table are used as place holders for the actual values of the JMS Listener you are troubleshooting.

Important: If you don’t have any of items in the table, stop now – there’s nothing you can do. If you have only the first four, you can still perform some diagnostic steps.

## Is my JMS Listener running?

Log in to the xBRMS database and run the following queries a few times:

SELECT count(\*) FROM [XBRMS-DB-NAME].[rdr].[Event]

SELECT count(\*) FROM [XBRMS-DB-NAME].[gxp].[BusinessEvent]

SELECT count(\*) FROM [XBRMS-DB-NAME].[gxp].[EntitlementStatus]

### All counts increase over time

Your JMS Listener is up running.

### None of the counts increase over time

First, verify that your listener is in fact running. Check that the server (physical or virtual) that hosts your JMS Listener is turned on. You can first try “pinging” the server to see if it’s available over the network. From any Windows or Linux machine on the network type:

ping JMSLIST-SERVER

If the ping command reports “destination host unreachable”, you will need to consult with an IT resource to restart the JMS Listener server or otherwise determine why it is not accessible over the network.

If the ping responds successfully, then the JMS Listener server is running but, for some reason, the JMS Listener service is not. Follow steps from section [What to do if my JMS Listener is not running](#_What_to_do_2) to start your JMS Listener.

Once you restart your JMS Listener, test it by running the above queries a few more times.

### BusinessEvent and/or EntitlementStatus tables’ counts don’t increase over time, but the Event’s table count increases

If the counts from the BusinessEvent and/or EntitlementStatus tables don’t increase, but you have verified that new events are being written to the Event table, your JMS Listener is running but no JMS messages are being written to the GXP-BROKER-TOPIC.

Refer to appropriate GxP trouble shooting guide to troubleshoot GxP to find out why.

### Event count doesn’t change but the BusinessEvent and/or EntitlementStatus tables’ counts increase over time

If the count from the Event table query doesn’t increase, but you see the number of events in either the BusinessEvent or EntitlementStatus tables increase, your JMS Listener is running but no JMS messages are being written to the XBRC-BROKER-TOPIC.

Refer to the [Diagnosing xBRC Problems](#_Diagnosing_xBRC_Problems) section of this guide to make sure your xBRC is healthy, its JMS Broker configuration is correct, and sending its JMS messages either upstream or to the SonicMQ.

## What to do if my JMS Listener is not running

To start the JMS Listener service, log into the JMS Listener server by running *putty* on a Windows machine or *ssh* on a Linux machine. For example, from Linux, can use:

ssh JMSLIST-USER@JMSLIST-SERVER

Windows *putty* users will enter the JMS Listener IP address and the login name using interactive screens. Regardless, when prompted for a password, type in the appropriate password for your JMS Listener login user. At this point, you will be presented with a Linux prompt (the JMS Listener runs on a Red Hat 5.6 compatible operating system). To verify the status of the JMS Listener service, type in this command:

#sudo /etc/init.d/jmslistener status

Do not type the “#” character – it is the Linux prompt indicating that you are running as a privileged user.

If you are prompted for a password, type in your login password again. If the JMS Listener is not running (as you would expect), the command should return:

No pidfile present

If it *is* running (but somehow, not responding to the /status URL), the command will return:

Service running with pid: ####

If the JMS Listener is not running, you can start it with:

#sudo /etc/init.d/jmslistener start

If it is running, but not responsive, make sure that there is only one instance of the service running:

#sudo ps ax | grep jms

If you see more than one instance running, stop them all using this command:

#kill ####

Where #### is the appropriate process number.

Now start the listener:

#sudo /etc/init.d/jmslistener start

## Is my JMS Listener configured correctly?

To make sure your JMS Listener is configured correctly log in to its server and make sure that values in the properties.xml file are correct.

Log into the JMS Listener server by running *putty* on a Windows machine or *ssh* on a Linux machine. For example, from Linux, can use:

ssh JMSLIST-USER@JMSLIST-SERVER

Edit the properties.xml file with your favorite editor:

# vi /usr/share/jmslistener/properties.xml

Verify that the entries in the properties.xml file are correct and change them when appropriate:

1. <entry key="gxpSource">**GXP-BROKER-SERVER**:**GXP-BROKER-PORT**;**GXP-BROKER-TOPIC**;**GXP-BROKER-USER**;**GXP-BROKER-PASSWORD**</entry>
2. <entry key="gxpDestination">jdbc:jtds:sqlserver://**XBRMS-DB-IP**:**XBRMS-DB-PASSWORD**/**XBRMS-DB-NAME**;user=**XBRMS-DB-USER**;password=**XBRMS-DB-PASSWORD**</entry>
3. <entry key="xbrcSource">**XBRC-BROKER-SERVER**:**XBRC-BROKER-PORT**;**XBRC-BROKER-TOPIC**;**XBRC-BROKER-USER**;**XBRC-BROKER-PASSWORD**</entry>
4. <entry key="xbrcDestination">jdbc:jtds:sqlserver://**XBRMS-DB-IP**:**XBRMS-DB-PORT**;databaseName=**XBRMS-DB-NAME**;user=**XBRMS-DB-USER**;password=**XBRMS-DB-PASSWORD**</entry>
5. <entry key="IDMSURL">http://**IDMS-SERVER**:**IDMS-PORT**/IDMS/</entry>
6. <entry key="GXPBookingCallbackURL">[http://**GXP-SERVER**:**GXP-PORT**/gxp-web/services/booking/xpass</entry](http://nl-flfa-00071.wdw.disney.com:8080/gxp-web/services/booking/xpass%3c/entry)>

Restart your JMS Listener if you changed any of its configuration values:

# vi /usr/share/jmslistener restart