



[HTTP://REMEDYLEGACY.COM](http://remedylegacy.com)

APL LOAD BALANCER MONITOR v 2.1

Contents

Introduction.....	1
Monitoring Modes.....	1
Load Balancer Health Check.....	1
Deploying.....	1
Initial Setup.....	2
Configure the Monitor.....	2
Install the Monitor as a Service (Windows).....	4
Run the tool manually (Unix/Linux).....	4
Release Notes.....	5

INTRODUCTION

This program is used to monitor the availability of either a Remedy AR System server or a Remedy Mid-Tier instance. Use this tool in conjunction with a load balancer's health check feature to route traffic to responsive servers.

Monitoring just a Remedy server's TCP port or the Mid-Tier's web server port is an indication that a process is allowing TCP connections. Such a connection does not ensure that the system is responsive in a way that is useful to users or integrated systems. Many Remedy admins have experienced situations where the Remedy server is up and technically running but is "hung" and not processing responses to connected clients, often leading to timeout errors. APL Load Balancer Monitor (APL LBM) goes beyond just a process allowing TCP connections and "works" the Remedy server to make sure the system is in a useful state.

MONITORING MODES

APL LBM has two different monitoring modes, Remedy server and Mid-Tier. When configured to monitor a Remedy server APL LBM logs into the Remedy server and executes a service call against the Service Form 'APL:LBMonitor' populating field 'Process' (700000001) with the value 'ExerciseServer'. This guide is configured out of the box to perform a simple query of the user form checking to ensure that there are records in the table. This filter guide can be customized to include any steps that you desire to confirm that your Remedy server is 'up'. This process allows for a customizable monitoring that works in your environment and that AR System Server is fully processing system requests. When configured to monitor a Mid-Tier server APL LBM connects to the Mid-Tier via URL accessing the 'APL:LBMonitor' form to verify the Mid-Tier is processing form requests (and in turn system requests). In either mode when the routine runs without error, it means the target server is responding normally and that traffic should be routed to it.

NOTIFICATION

The system, if configured properly, will send out notifications to the emails specified when the port is taken up or down. The notification system uses smtp for no authentication, SSL, or TLS. You can specify the to email can be either a single email address or a list that's either comma or semicolon separated list of addresses.

LOAD BALANCER HEALTH CHECK

When APL LBM determines the monitored process is responsive, a configurable TCP port is opened by the monitor. You configure the load balancer's health check to monitor this port as an indicator that the system is available for network traffic. If for any reason the server routine fails the port is closed and the load balancer should stop sending network traffic to that endpoint.

DEPLOYING

Most often APL LBM is deployed on the server that is being monitored however it can be run from any server in the environment. A single instance of this tool can only monitor either a Remedy server or a Mid-Tier server. Multiple instances of APL LBM can be run on the same server if there is a need to monitor both a Remedy server and Mid-Tier from the same server (for example a server with both Remedy server and Mid-Tier are installed), you will need to ensure that multiple instances are using different ports to ensure that each monitor can open its own port.

INITIAL SETUP

The setup of this utility requires importing conf/APL_LBMonitor.def onto the Remedy server that's going to be monitored, this is required no matter if you are monitoring Remedy or Mid-Tier as the monitor tool uses the form in question in both cases. Then copy the files to the server, adjusting the configuration and installing the service (Windows) or setting it up to auto start in Unix/Linux.

CONFIGURE THE MONITOR

1. Unzip the APL LBM zip files to a suitable place on the server

Example: D:\Utilities\APL\APLLBMonitor

2. Open conf\config.properties file configure the following:

Parameter	Description	Example Values
monitor	Monitor either a Remedy server or Mid-Tier	remedy, midtier
serviceForm	This is the form that the tool uses to manage 'up'	APL:LBMonitor
monitorPort	The port the LB's health check will monitor	3636
checkIntervalWhenUp	How often in seconds Remedy/Mid-Tier server will be checked when APL LBM is in an "up" status	20
checkIntervalWhenDown	How often in seconds Remedy/Mid-Tier server will be checked when APL LBM is in a "down" status	5
downThreshold	How many failed checks before APL LBM considers Remedy/Mid-Tier in a down status and closes the monitorPort	3
server	Remedy server to be monitored. Not required if monitoring Mid-Tier.	rmdyapp01
tcpPort	Remedy Server TCP port if needed. Not required if monitoring Mid-Tier.	
user	Username of account that will login to Remedy and perform the check. Must have access to 'APL:LBMonitor' form.	Demo
pass	Password of the account that will login to Remedy and perform the check.	
useRAS	Tells APL LBM to use Remedy	true, false

	Application Service account. If set, and the RASValue is not already set, the user/password provided must be an admin account to be able to access the form to get the RAS Value	
RASValue	Will automatically be set the first time APL LBM is run with useRAS = true, if you want to define manually, you can grab the value out of the ar.cfg/ar.conf for parameter 'Remedy-App-Service-Password'	
url	Used when monitor = midtier. Includes the URL path to the Mid-Tier server and AR Server that will be used to load the APL:LBMonitor form	https://rmdyweb01:8080/arsys/forms/rmdyapp01
smtpServer	This is the SMTP server that you want used for sending notifications	smtp.gmail.com
smtpPort	The port you use to connect to the SMTP Server	25 (non secure), 465 (SSL), 587 (TLS)
authType	This will be 'none', 'ssl', or 'tls' depending on your needs	
smtpUser	If authType is anything other than 'none', you will need to provide a username and password to authenticate	
smtpPass		
fromEmail	This would be the email address you want the notification to come from, you will need to have permission to send on behalf of this user	remedy@company.com
fromEmailName	If populated, this would be the display name used by the email address	Remedy Admin
toEmail	This is one or many email addresses you wish to send to either , or ; separated	admin1@company.com , admin2@company.com ; admin3@company.com

INSTALL THE MONITOR AS A SERVICE (WINDOWS)

In order to run this tool as a service in Windows, we are utilizing the Apache Procrun tool (<https://commons.apache.org/proper/commons-daemon/procrun.html>). Assuming your JAVA_HOME or JRE_HOME environment variables are set properly, all you would need to do to get this functional would be to run the bin\service.bat file with an 'install' parameter. If you don't want the service named 'APLLBMonitor', then you can specify an additional parameter of the name you want it to be named

RUN THE TOOL MANUALLY (UNIX/LINUX)

If you won't be running this under Windows, you can change the directory structure a bit to make things a bit easier. You may want to move the config file into the same directory as the main jar, or you can use a -c parameter to the program to specify the location of the config file. If running manually, your command execution may look something like this

```
Java -cp ./lib/APLLBMonitor.jar;./lib/arapi81_build001.jar;./lib/arutil81_build001.jar;./lib/log4j-1.2.14.jar;./lib/javax.mail.jar com.apl.Daemon -c ./conf/config.properties
```

This command line assumes you left everything in their original directories and are sitting at the parent folder of both lib and conf. This will produce a log file in the logs directory (assuming that you haven't changed the log4j section of the config.properties). If you wish the log to appear somewhere else, you can specify a fully qualified path, or a relative path in the config.properties file

RELEASE NOTES

2.1 – Mar 16, 2021

- Found and fixed a potential bug with the checker thread not closing out and causing a resource issue

2.0 – Nov 24, 2020

- The tool has been updated to be able to be run with the Apache RunProc wrapper instead of previous capability
- Capability was added to allow for notification when the port is taken up or down

1.3 – Jan 26, 2019

- Updated the code, documentation and config to allow for a user on Linux to start the tool in the way they want
- Modified the default logging in the config.properties to not log AR Exceptions to the log as they are already being handled in the code
- Put it so that the JAR doesn't bundle the api again so that the api version can be managed independently of the tool version

1.2 – Dec 15th, 2018

- Modified program to use Log4J so that if you aren't using the Wrapper in Windows, you still can control the size and number of the output log

1.1 – June 12, 2017

- Fixed problem with the services inability to find the config file

1.0 – June 10, 2017

- Initial public pre-release made available for download