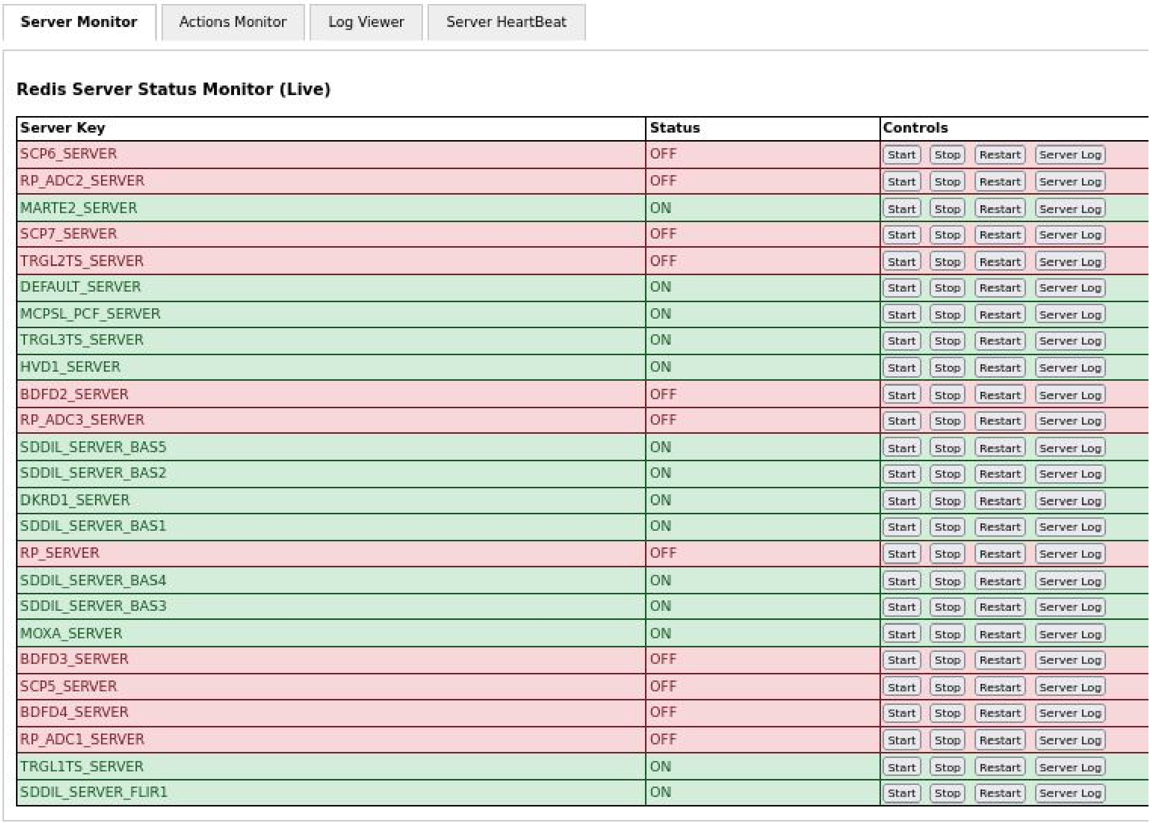
MDSplus Redis Dispatcher WebMonitor – Short Reference Manual

This web application provides real-time visibility and control over distributed actions and servers using a Redis-backed architecture. It is intended for system operators, developers, or engineers overseeing MDSPlus Redis Dispatcher automated processes.

Main Tabs Overview

Tab	Purpose
Server Monitor	View the real-time status (ON/OFF) of all active servers.
Actions Monitor	Monitor action status by tree/shot. Dispatch or abort actions.
Log Viewer	Submit log messages manually (for testing) and stream live logs from the server log file.
Server HeartBeat	View last heartbeat activity timestamp for each server.

Server Monitor



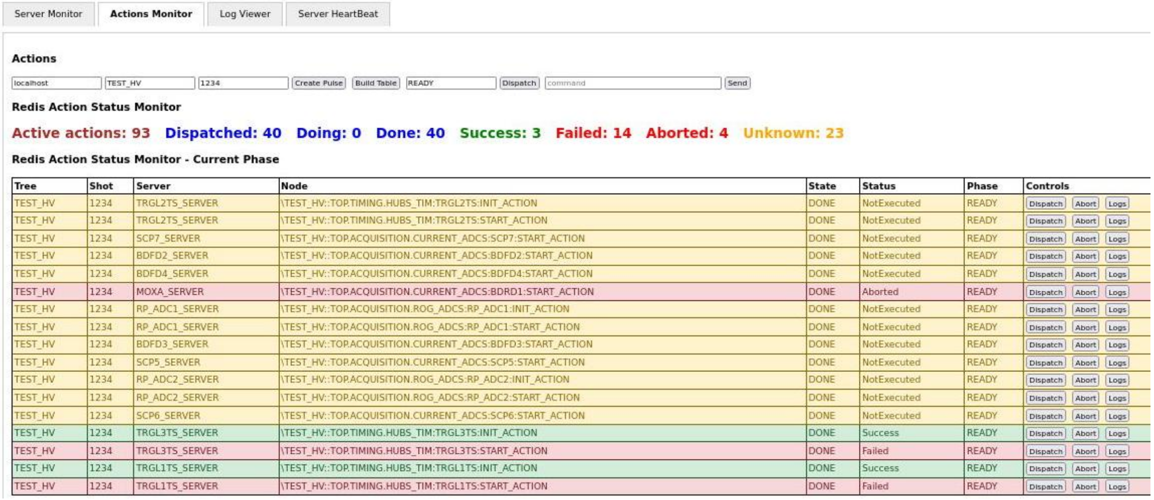
Redis Server Status Monitor (Live)

Server Key	Status	Controls
SCP6_SERVER	OFF	Start Stop Restart Server Log
RP_ADC2_SERVER	OFF	Start Stop Restart Server Log
MARTE2_SERVER	ON	Start Stop Restart Server Log
SCP7_SERVER	OFF	Start Stop Restart Server Log
TRGL2TS_SERVER	OFF	Start Stop Restart Server Log
DEFAULT_SERVER	ON	Start Stop Restart Server Log
MCPSL_PCF_SERVER	ON	Start Stop Restart Server Log
TRGL3TS_SERVER	ON	Start Stop Restart Server Log
HVD1_SERVER	ON	Start Stop Restart Server Log
BDFD2_SERVER	OFF	Start Stop Restart Server Log
RP_ADC3_SERVER	OFF	Start Stop Restart Server Log
SDDIL_SERVER_BAS5	ON	Start Stop Restart Server Log
SDDIL_SERVER_BAS2	ON	Start Stop Restart Server Log
DKRD1_SERVER	ON	Start Stop Restart Server Log
SDDIL_SERVER_BAS1	ON	Start Stop Restart Server Log
RP_SERVER	OFF	Start Stop Restart Server Log
SDDIL_SERVER_BAS4	ON	Start Stop Restart Server Log
SDDIL_SERVER_BAS3	ON	Start Stop Restart Server Log
MOXA_SERVER	ON	Start Stop Restart Server Log
BDFD3_SERVER	OFF	Start Stop Restart Server Log
SCP5_SERVER	OFF	Start Stop Restart Server Log
BDFD4_SERVER	OFF	Start Stop Restart Server Log
RP_ADC1_SERVER	OFF	Start Stop Restart Server Log
TRGL1TS_SERVER	ON	Start Stop Restart Server Log
SDDIL_SERVER_FLIR1	ON	Start Stop Restart Server Log

- Live status of all servers registered in Redis (ACTION_SERVER_ACTIVE).
- Each row includes: Server key, Status (ON/OFF)

- Each row includes: Control buttons: START, STOP, RESTART, SERVER_LOG
- Auto-updates the server information.

Actions Monitor



Actions

localhost | TEST_HV | 1234 | READY | command

Redis Action Status Monitor

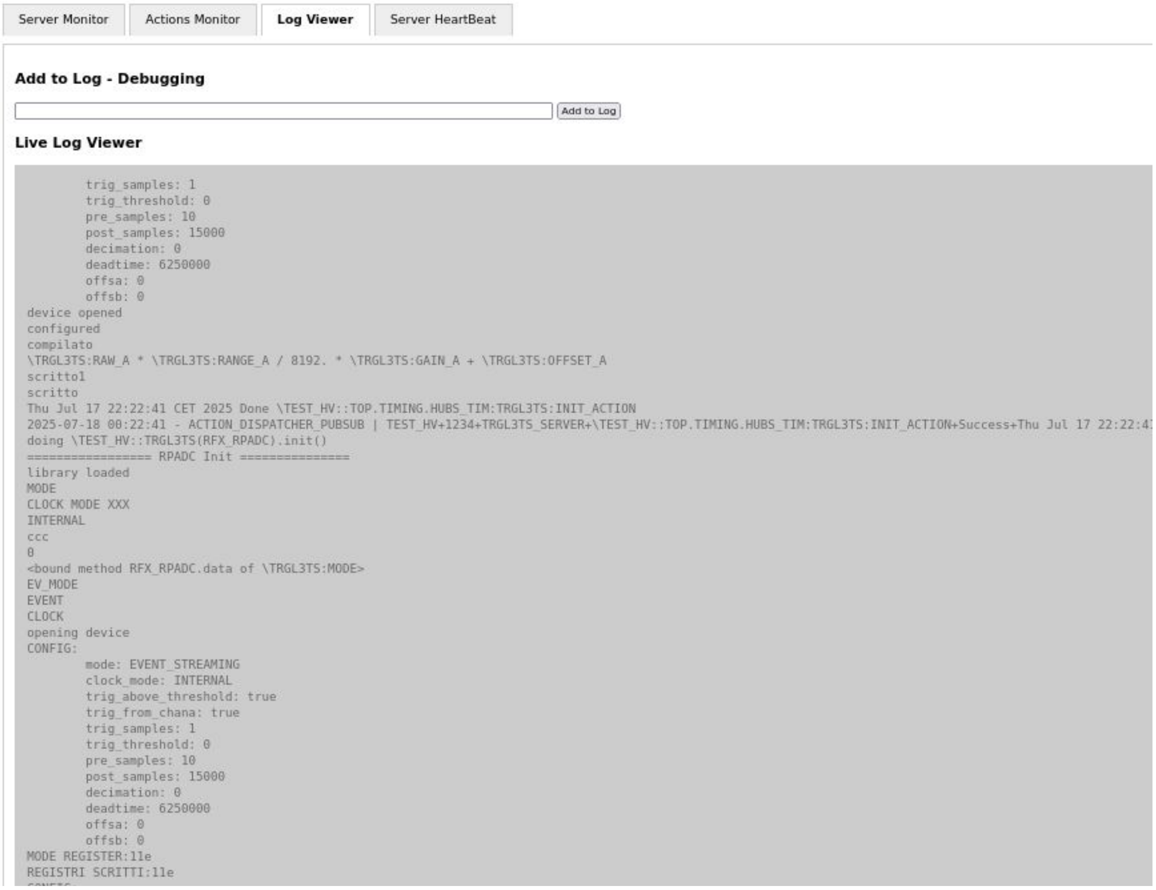
Active actions: 93 Dispatched: 40 Doing: 0 Done: 40 Success: 3 Failed: 14 Aborted: 4 Unknown: 23

Redis Action Status Monitor - Current Phase

Tree	Shot	Server	Node	State	Status	Phase	Controls
TEST_HV	1234	TRGL2TS_SERVER	\TEST_HV::TOP.TIMING.HUBS_TIM:TRGL2TS:INIT_ACTION	DONE	NotExecuted	READY	Dispatch Abort Logs
TEST_HV	1234	TRGL2TS_SERVER	\TEST_HV::TOP.TIMING.HUBS_TIM:TRGL2TS:START_ACTION	DONE	NotExecuted	READY	Dispatch Abort Logs
TEST_HV	1234	SCP7_SERVER	\TEST_HV::TOP.ACQUISITION.CURRENT_ADCS:SCP7:START_ACTION	DONE	NotExecuted	READY	Dispatch Abort Logs
TEST_HV	1234	BDFD2_SERVER	\TEST_HV::TOP.ACQUISITION.CURRENT_ADCS:BDFD2:START_ACTION	DONE	NotExecuted	READY	Dispatch Abort Logs
TEST_HV	1234	BDFD4_SERVER	\TEST_HV::TOP.ACQUISITION.CURRENT_ADCS:BDFD4:START_ACTION	DONE	NotExecuted	READY	Dispatch Abort Logs
TEST_HV	1234	MOXA_SERVER	\TEST_HV::TOP.ACQUISITION.CURRENT_ADCS:BDRD1:START_ACTION	DONE	Aborted	READY	Dispatch Abort Logs
TEST_HV	1234	RP_ADC1_SERVER	\TEST_HV::TOP.ACQUISITION.ROG_ADCS:RP_ADC1:INIT_ACTION	DONE	NotExecuted	READY	Dispatch Abort Logs
TEST_HV	1234	RP_ADC1_SERVER	\TEST_HV::TOP.ACQUISITION.ROG_ADCS:RP_ADC1:START_ACTION	DONE	NotExecuted	READY	Dispatch Abort Logs
TEST_HV	1234	BDFD3_SERVER	\TEST_HV::TOP.ACQUISITION.CURRENT_ADCS:BDFD3:START_ACTION	DONE	NotExecuted	READY	Dispatch Abort Logs
TEST_HV	1234	SCP5_SERVER	\TEST_HV::TOP.ACQUISITION.CURRENT_ADCS:SCP5:START_ACTION	DONE	NotExecuted	READY	Dispatch Abort Logs
TEST_HV	1234	RP_ADC2_SERVER	\TEST_HV::TOP.ACQUISITION.ROG_ADCS:RP_ADC2:INIT_ACTION	DONE	NotExecuted	READY	Dispatch Abort Logs
TEST_HV	1234	RP_ADC2_SERVER	\TEST_HV::TOP.ACQUISITION.ROG_ADCS:RP_ADC2:START_ACTION	DONE	NotExecuted	READY	Dispatch Abort Logs
TEST_HV	1234	SCP6_SERVER	\TEST_HV::TOP.ACQUISITION.CURRENT_ADCS:SCP6:START_ACTION	DONE	NotExecuted	READY	Dispatch Abort Logs
TEST_HV	1234	TRGL3TS_SERVER	\TEST_HV::TOP.TIMING.HUBS_TIM:TRGL3TS:INIT_ACTION	DONE	Success	READY	Dispatch Abort Logs
TEST_HV	1234	TRGL3TS_SERVER	\TEST_HV::TOP.TIMING.HUBS_TIM:TRGL3TS:START_ACTION	DONE	Failed	READY	Dispatch Abort Logs
TEST_HV	1234	TRGL1TS_SERVER	\TEST_HV::TOP.TIMING.HUBS_TIM:TRGL1TS:INIT_ACTION	DONE	Success	READY	Dispatch Abort Logs
TEST_HV	1234	TRGL1TS_SERVER	\TEST_HV::TOP.TIMING.HUBS_TIM:TRGL1TS:START_ACTION	DONE	Failed	READY	Dispatch Abort Logs

- Shows two tables: Active Actions (based on CURRENT_PHASE) and All Actions (regardless of phase).
- Each row includes the information: Tree / Shot / Server / Node / State / Status / Phase
- Each row also permits the following actions DISPATCH, ABORT, LOGS buttons.
- Includes a dispatcher form for pulse creation, table builds, and custom commands.
- Color-coded rows (Blue: Not Dispatched, Green: Success, Red: Failed or Aborted, Orange: In progress).
- A Live summary shows total and categorized status counts.

Log Viewer



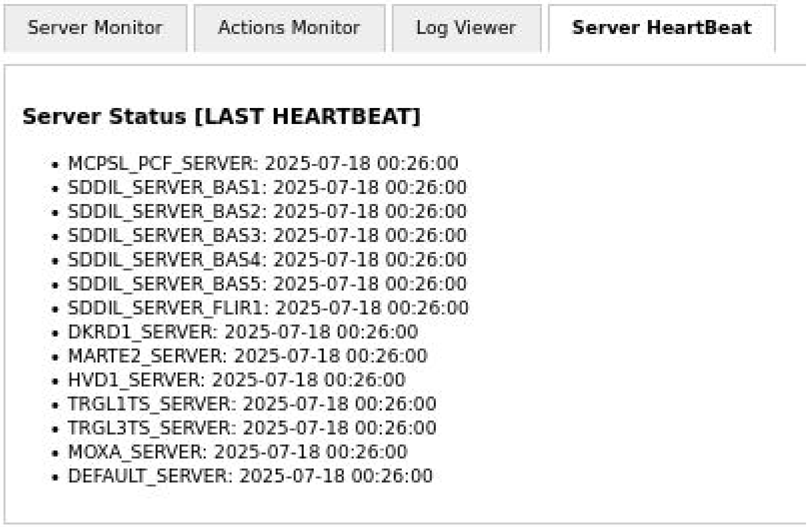
Add to Log - Debugging

Live Log Viewer

```
        trig_samples: 1
        trig_threshold: 0
        pre_samples: 10
        post_samples: 15000
        decimation: 0
        deadtime: 6250000
        offsa: 0
        offsb: 0
device opened
configured
compilato
\TRGL3TS:RAW_A * \TRGL3TS:RANGE_A / 8192. * \TRGL3TS:GAIN_A + \TRGL3TS:OFFSET_A
scrittol
scritto
Thu Jul 17 22:22:41 CET 2025 Done \TEST_HV::TOP.TIMING.HUBS_TIM:TRGL3TS:INIT_ACTION
2025-07-18 00:22:41 - ACTION_DISPATCHER_PUBSUB | TEST_HV+1234+TRGL3TS_SERVER+\TEST_HV::TOP.TIMING.HUBS_TIM:TRGL3TS:INIT_ACTION+Success+Thu Jul 17 22:22:4
doing \TEST_HV::TRGL3TS(RFX_RPADC).init()
================= RPADC Init ===============
library loaded
MODE
CLOCK MODE XXX
INTERNAL
ccc
0
<bound method RFX_RPADC.data of \TRGL3TS:MODE>
EV_MODE
EVENT
CLOCK
opening device
CONFIG:
        mode: EVENT_STREAMING
        clock_mode: INTERNAL
        trig_above_threshold: true
        trig_from_chana: true
        trig_samples: 1
        trig_threshold: 0
        pre_samples: 10
        post_samples: 15000
        decimation: 0
        deadtime: 6250000
        offsa: 0
        offsb: 0
MODE REGISTER:11e
REGISTRI SCRITTI:11e
```

- Submit custom log messages to the application log.
- Live stream of redis_pubsub.log file with auto-scrolling.

Server HeartBeat



Server Status [LAST HEARTBEAT]

- MCPSL_PCF_SERVER: 2025-07-18 00:26:00
- SDDIL_SERVER_BAS1: 2025-07-18 00:26:00
- SDDIL_SERVER_BAS2: 2025-07-18 00:26:00
- SDDIL_SERVER_BAS3: 2025-07-18 00:26:00
- SDDIL_SERVER_BAS4: 2025-07-18 00:26:00
- SDDIL_SERVER_BAS5: 2025-07-18 00:26:00
- SDDIL_SERVER_FLIR1: 2025-07-18 00:26:00
- DKRD1_SERVER: 2025-07-18 00:26:00
- MARTE2_SERVER: 2025-07-18 00:26:00
- HVD1_SERVER: 2025-07-18 00:26:00
- TRGL1TS_SERVER: 2025-07-18 00:26:00
- TRGL3TS_SERVER: 2025-07-18 00:26:00
- MOXA_SERVER: 2025-07-18 00:26:00
- DEFAULT_SERVER: 2025-07-18 00:26:00

- Displays last known heartbeat timestamp for each server via pub/sub.

- Updates on receiving messages from ACTION_SERVER_PUBSUB:* channels.

Usage Tips

- Use DISPATCH only after create_pulse and build_tables have been run.
- Logs can help debug Redis messaging or command execution failures.
- Monitor the Active Actions table for current work-in-progress.
- The heartbeat tab can help diagnose the active list of servers being polled.

Starting the Web Server

To start the Redis Dispatcher WebMonitor, ensure the necessary dependencies are installed using pip:

```
pip install gunicorn gevent
```

You can launch the web server using the following Gunicorn command:

```
gunicorn -w 5 -k gevent --keep-alive 3600 --timeout 0 -b 0.0.0.0:5000 dispatcher_webmonitor:app
```

Explanation of the options:

- -w 5 — Start 5 worker processes.
- -k gevent — Use the Gevent worker class for asynchronous support.
- --keep-alive 3600 — Keep connections alive for up to 3600 seconds.
- --timeout 0 — Disable worker timeout (workers won't be killed for taking too long).
- -b 0.0.0.0:5000 — Bind the server to all interfaces on port 5000.
- dispatcher_webmonitor:app — The Python module and Flask app instance to serve.