

---

# Software Authority Protocol

TCP/IP Protocol for controlling endpoint routers

Fred Gleason

## Table of Contents

Overview .....	2
Protocol Command Syntax .....	2
Protocol Response Syntax .....	2
Booleans .....	3
Connection Management .....	3
Exit .....	3
Login .....	3
Quit .....	4
Configuration Discovery .....	4
List Destination Names [DestNames] .....	4
List Router Names [RouterNames] .....	5
List Snapshots [SnapShots] .....	5
List Source Names [SourceNames] .....	6
State Discovery .....	7
Get Crosspoint Route State [RouteStat] .....	7
Get General Purpose Input (GPI) State [GPISat] .....	8
Get General Purpose Output (GPO) State [GPOStat] .....	8
Commands .....	9
Set Snapshot [ActivateSnap / ActivateScene] .....	9
Set Crosspoint Route [ActivateRoute] .....	10
Set General Purpose Input (GPI) State [TriggerGPI] .....	10
Set General Purpose Output (GPO) State [TriggerGPO] .....	11
Extended Protocol Messages .....	12
Mask <b>GPISat</b> Update Messages .....	12
Mask <b>GPOStat</b> Update Messages .....	12
Mask <b>RouteStat</b> Update Messages .....	12
Mask All Update Messages .....	13

## Overview

Messages to the Software Authority routing service are by means of a TCP SOCK\_STREAM connection to TCP port 9500 on the host server.

## Protocol Command Syntax

Commands from a control client to the service have the following general syntax:

*cmd-phrase* [*arg*] [. . .]**CR/LF**

*cmd-phrase*                    A single phrase, consisting of one or more "camel-case" words; containing no whitespace, case-insensitive.

*arg*                                Zero or more arguments, delimited by a **space** character (ASCII 32).

**CR/LF**                            The ASCII character **CR** (13) followed by **LF** (10).

## Protocol Response Syntax

Responses from the service to the control client take one of two possible forms:

### State Updates

Single line message communicating a change of state, with the syntax:

*resp-phrase* [*arg*] [. . .]**CR/LF**

*resp-phrase*                    A single phrase, consisting of one or more "camel-case" words; containing no whitespace.

*arg*                                Zero or more arguments, delimited by a **space** character (ASCII 32).

**CR/LF**                            The ASCII character **CR** (13) followed by **LF** (10).

### Lists

Multi-line messages containing a list of multiple attributes. Lists are only sent in response to a command requesting them, and have the general format:

```
Begin cmd-phrase [- cmd-arg]CR/LF
    param1HTparam2HT[...]CR/LF
End cmd-phrase [- cmd-arg]CR/LF
```

Where:

*cmd-phrase*            The command phrase issued by the client, consisting of one or more "camel-case" words; containing no whitespace.

*cmd-arg*              An optional argument, provided by the command from the client.

**HT**                    The ASCII character **HT** [horizontal tab] (9).

**CR/LF**                The ASCII character **CR** (13) followed by **LF** (10).

A Begin/End block contains zero or more lines, each line consisting of HT delimited records.

## Booleans

Those commands that require and/or return boolean values do so by using the strings **True** or **False**.

## Connection Management

Messages for managing connections to the service.

### Exit

**Exit**

Drop the TCP connection and end the session.

### Login

**Login** *user-name password*

Authenticate to the service.

## Note

On the Drouter system, the **Login** command does nothing, and will accept any combination of *user-name* and *password* as being valid. It is provided strictly for compatibility with clients designed to use PathFinder.

## Quit

### Quit

Drop the TCP connection and end the session.

## Configuration Discovery

Messages for interrogating the system about its configuration. These commands all return list-type responses.

## List Destination Names [DestNames]

**DestNames** *router-num*

Return a list of destination endpoints on the specified router.

Returns:

```
Begin DestNames - router-num
    endpt-num short-name long-name node-addr node-name node-slot
End DestNames - router-num
```

Example:

```
>>DestNames 2
Begin DestNames - 2
  1  Mackie 3/4  Mackie 3/4 ON FRED-XNODE  172.30.4.211  FRED-XNODE  1
  2  Mackie 5/6  Mackie 5/6 ON FRED-XNODE  172.30.4.211  FRED-XNODE  2
  3  Delta 1/2   Delta 1/2 ON FRED-XNODE  172.30.4.211  FRED-XNODE  3
  9  PGM 4      PGM 4 ON RDVIRT-SERV  172.30.4.212  RDVIRT-SERV  1
 10 PGM 3      PGM 3 ON RDVIRT-SERV  172.30.4.212  RDVIRT-SERV  2
```

End DestNames - 2

## List Router Names [RouterNames]

### **RouterNames**

Return a list of routers configured on this system.

Returns:

```
Begin RouterNames
    router-num router-name
End RouterNames
```

Example:

```
>>RouterNames
Begin RouterNames
    1 RFA-ALL
    2 Fred-Test
    3 Fred-GPIO-Test
    4 RFA_AIR_MC
    7 Rivendell-Test
    9 GPIO
    11 RFA_AIRPLAYS_BLUE
    12 RFA_AIRPLAYS_GOLD
End RouterNames
```

## List Snapshots [SnapShots]

**SnapShots** *router-num*

Return a list of snapshots on the specified router.

Returns:

```
Begin SnapshotNames - router-num
    snapshot-name
End SourceNames - router-num
```

Example:

```
>>SnapshotNames 4
Begin SnapshotNames - 4
    BUR from FEED
    BUR from Studio 11
    TIB from MC
End SnapshotNames - 4
```

## List Source Names [SourceNames]

**SourceNames** *router-num*

Return a list of source endpoints on the specified router.

Returns:

```
Begin SourceNames - router-num
    endpt-num short-name long-name node-addr node-name node-slot src-num stream-addr
End SourceNames - router-num
```

Example:

```
>>SourceNames 2
Begin SourceNames - 2
  1  Mackie Main  Mackie Main ON FRED-XNODE  172.30.4.211  FRED-XNODE  1  30001  239.192.117.49
  2  SRC 2      SRC 2 ON FRED-XNODE  172.30.4.211  FRED-XNODE  2  30002  239.192.117.50
  9  Rivenberry  Rivenberry ON RDVIRT-SERV  172.30.4.212  RDVIRT-SERV  1  30301  239.192.118.93
 10 SRC 2      SRC 2 ON RDVIRT-SERV  172.30.4.212  RDVIRT-SERV  2  30302  239.192.118.94
```

End SourceNames - 2

## State Discovery

Messages for interrogating the system about its current state. These commands all return single-line style responses.

### Get Crosspoint Route State [RouteStat]

**RouteStat** *router-num* [*endpt-num*]

If the *endpt-num* argument is omitted, the system will send RouteStat messages for *all* of the destinations that belong to the specified *router-num*.

Interrogate one or more destinations for their connected source. Returns messages of the following form:

RouteStat *router-num* *dest-endpt-num* *src-endpt-num* *lock-active*

#### Note

The *lock-active* field will *always* be "False" on Drouter as Drouter does not support the notion of locking routes.

A returned value of "0" for *src-endpt-num* indicates that the respective destination is either disconnected, or connected to a source that is not a member of the specified *router-num*.

Example (single destination):

```
>>RouteStat 2 2
RouteStat 2 2 17 False
```

Example (entire router):

```
>>RouteStat 2
RouteStat 2 1 17 False
RouteStat 2 2 2 False
RouteStat 2 3 0 False
RouteStat 2 9 1 False
```

```
RouteStat 2 10 1 False
```

## Get General Purpose Input (GPI) State [GPISat]

**GPISat** *router-num* [*endpt-num*]

If the *endpt-num* argument is omitted, the system will send GPISat messages for *all* of the sources that belong to the specified *router-num*.

Interrogate one or more sources for their GPI state. Returns messages of the following form:

```
GPISat router-num endpt-num state-str
```

The returned *state-str* will be a five character string indicating the low (l) or high (h) state of the respective GPI source.

### Note

An error will be returned if the specified *router-num* is not a GPIO router.

Example (single source):

```
>>GPISat 3 11
GPISat 3 11 lhlhh
```

Example (entire router):

```
>>GPISat 3
GPISat 3 1 lhlhh
GPISat 3 2 hhhhh
GPISat 3 3 hlhhh
GPISat 3 4 hhhhh
GPISat 3 5 hhhhh
GPISat 3 6 hhhhh
```

## Get General Purpose Output (GPO) State [GPOStat]

**GPOStat** *router-num* [*endpt-num*]



If the *endpt-num* argument is omitted, the system will send `GPOStat` messages for *all* of the destinations that belong to the specified *router-num*.

Interrogate one or more sources for their GPO state. Returns messages of the following form:

`GPOStat router-num endpt-num state-str`

The returned *state-str* will be a five character string indicating the low (l) or high (h) state of the respective GPO destination.

### Note

An error will be returned if the specified *router-num* is not a GPIO router.

Example (single source):

```
>>GPOStat 3 11
GPOStat 3 11 lhlhh
```

Example (entire router):

```
>>GPOStat 3
GPOStat 3 1 lhlhh
GPOStat 3 2 hhhhh
GPOStat 3 3 hlhhh
GPOStat 3 4 hhhhh
GPOStat 3 5 hhhhh
GPOStat 3 6 hhhhh
```

## Commands

Messages for actively changing the state of the system.

### Set Snapshot [ActivateSnap / ActivateScene]

**ActivateSnap** *router-num snapshot-name*

The system will respond with zero or more **RouteStat** messages to reflect changed crosspoint states.

**Note**

**ActivateScene** is merely a synonym for **ActivateSnap**. The syntax and operation of the two commands are identical.

Example:

```
>>ActivateSnap 4 MAN from MC
RouteStat 4 3 9 False
RouteStat 7 3 0 False
RouteStat 1 5045 1953 False
```

## Set Crosspoint Route [ActivateRoute]

**ActivateRoute** *router-num dest-endpt-num src-endpt-num*

The system will respond with zero or more **RouteStat** messages to reflect changed crosspoint state.

Example:

```
>>ActivateRoute 2 3 9
RouteStat 2 3 9 False
RouteStat 7 3 0 False
RouteStat 1 5045 1953 False
```

## Set General Purpose Input (GPI) State [TriggerGPI]

**TriggerGPI** *router-num src-endpt-num state-str [duration]*

The specified *state-str* should be a five character string indicating the state to which to set the specified GPI endpoint. Valid characters are:

- h** Set the corresponding line to the HIGH state.
- l** Set the corresponding line to the LOW state.
- x** Leave the corresponding line state UNCHANGED.

**Caution**

The *duration* parameter exists solely for compatibility with PathFinder. Drouter does not support setting momentary GPI states, thus, any *duration* parameter given on Drouter will be silently ignored.

**Note**

An error will be returned if the specified *router-num* is not a GPIO router.

The system will respond with zero or more **GPIStat** messages to reflect changed GPI state. Only virtual GPIO devices can have their GPI state set remotely; attempts to do so to physical GPI devices will be silently ignored.

Example:

```
>>TriggerGPI 3 11 xxlxx
GPIStat 3 11 hhlhh
```

## Set General Purpose Output (GPO) State [TriggerGPO]

**TriggerGPO** *router-num dest-endpt-num state-str [duration]*

The specified *state-str* should be a five character string indicating the state to which to set the specified GPO endpoint. Valid characters are:

- h** Set the corresponding line to the HIGH state.
- l** Set the corresponding line to the LOW state.
- x** Leave the corresponding line state UNCHANGED.

**Caution**

The *duration* parameter exists solely for compatibility with PathFinder. Drouter does not support setting momentary GPO states, thus, any *duration* parameter given on Drouter will be silently ignored.

**Note**

An error will be returned if the specified *router-num* is not a GPOO router.

The system will respond with zero or more **GPOStat** messages to reflect changed GPO state.

Example:

```
>>TriggerGPO 3 11 xxlxx  
GPOStat 3 11 hhlhh
```

## Extended Protocol Messages

### Caution

The messages described in this section are all "extended" inasmuch as they are Drouter-specific extensions to Software Authority Protocol. Hence, they are not present in non-Drouter implementations of the protocol!

## Mask GPIStat Update Messages

**DrouterMaskGPIStat True | False**

If set to **True**, this command will suppress any **GPIStat** response message(s) that would otherwise have been received on the connection upon which it was issued.

### Note

The **DrouterMaskGPIStat** command affects only the specific protocol connection in which it is used.

## Mask GPOStat Update Messages

**DrouterMaskGPOStat True | False**

If set to **True**, this command will suppress any **GPOStat** response message(s) that would otherwise have been received on the connection upon which it was issued.

### Note

The **DrouterMaskGPOStat** command affects only the specific protocol connection in which it is used.

## Mask RouteStat Update Messages

**DrouterMaskRouteStat True | False**

If set to **True**, this command will suppress any **RouteStat** response message(s) that would otherwise have been received on the connection upon which it was issued.

### Note

The **DrouterMaskRouteStat** command affects only the specific protocol connection in which it is used.

## Mask All Update Messages

**DrouterMaskStat** **True** | **False**

If set to **True**, this command will suppress all route state update messages --i.e. **GPIStat**, **GPOStat** and **RouteStat** messages -- that would otherwise have been received on the connection upon which it was issued.

### Note

The **DrouterMaskRouteStat** command affects only the specific protocol connection in which it is used.