STARKEEPER.IT

Voyager Application Server Protocol

Events, Methods and Workflow (TCP-IP)

Leonardo Orazi

05/05/2019

INDEX

1. Introduction

VOYAGER have an internal Application Server that allow external application to interact with it :

- receiving events
    - setup events
    - action events
    - error events
- send commands
    - setup cmd
    - action run
    - profile management

2. Connection

Clients connect to Voyager on TCP-IP port 5950. When multiple Voyager instances are running, each instance listens on successive port numbers (5951, 5952, ...). Max instance in the same PC is 3. Firewall must be opened to allow communications in the O.S.

VOYAGER allows multiple clients to establish connections simultaneously.

When a client establishes a connection, VOYAGER sends a version event messages to the client (see the events section). Notification messages are sent to all connected clients, answer to command only to relate client.

3. HeartBeat

Communication between Server/Client is under HeartBeat keep-alive system. If 15s passed without receiving valid data from client the server close the connection for inactivity. If you want to leave connection opened with server but you don't have data or command to send you must send a polling event each 5s to avoid connection closing, using a polling timer. Also if the server don't have valid data to send will use polling event each 5s to send to the client , in this way client know that server is running and connected and can manage (if needed) then closing itself.

Each communications valid received reset the inactivity timeout client side and server side, in this case the polling timer will be (must be) cleared and restarted. You must implements this polling procedure in your client.

4. Events

Event Notification messages are formatted as JSON objects. Each message is a single line of text terminated by CR LF.

Common attributes

All messages contain the following attributes in common:

Attribute	Type	Description
`Event`	String	the name of the event
`Timestamp`	number	the timestamp of the event in seconds from the epoch, including fractional seconds
`Host`	String	the hostname of the machine running VOYAGER
`Inst`	Integer	the VOYAGER instance number (1-based)

a) Version

Contains info about Voyager version

Attribute	Type	Description
`VOYVersion`	String	the version of Voyager
`VOYSubver`	String	the subversion of Voyager if present
`MsgVersion`	Integer	The numeric version of protocol implemented in this version of Voyager

Example:

```
{"Event":"Version","Timestamp":1550018143.66187,"Host":"hal9000","Inst":1,"VOYVe
rsion":"Release 2.0.14f - Built 2019-02-11","VOYSubver":"","MsgVersion":1}
```

b) Polling

Protocol Heartbeat. Send according HeartBeat paragraph.

Example:

```
{"Event":"Polling","Timestamp":1548806904.00159,"Host":"hal9000","Inst":1}
```

c) Signal

Used from server to send signal about something happen in Voyager, status changed, action started, error raised etc etc. Signals are sended in realtime.

Attribute	Type	Description
`Code`	Integer	The numeric index of Signal happen. See table below.

Code	Description

1	Autofocus Error
2	Remote Action RUN - Running Queue is empty
3	Remote Action RUN - SC ARRAY Autofocus all nodes
4	Remote Action RUN - Precise Pointing
5	Remote Action RUN - Autofocus
6	Remote Action RUN - SC ARRAY AutoFlat single node
7	Remote Action RUN - SC ARRAY Autofocus single node
8	Remote Action RUN - SC ARRAY Connect Setup all nodes
9	Remote Action RUN - SC ARRAY Disconnect Setup all nodes
10	Remote Action RUN - SC ARRAY Filter Change single node
11	Remote Action RUN - SC ARRAY Get Actual Filter single node
12	Remote Action RUN - SC ARRAY Focuser Move To single node
13	Remote Action RUN - SC ARRAY Focuser Offset single node
14	Remote Action RUN - SC ARRAY Rotator Move single node
15	Remote Action RUN - Setup Connect
16	Remote Action RUN - Setup Disconnect
18	Remote Action RUN - Camera Shot
19	Remote Action RUN - CCD Cooling
20	Remote Action RUN - Focuser Move To
21	Remote Action RUN - Focuser OffSet
22	Remote Action RUN - Rotator Goto
23	Remote Action RUN - AutoFlat
24	Remote Action RUN - Filter Change To
25	Remote Action RUN - Plate Solving Actual Location
26	Remote Action RUN - SC ARRAY Sequence
27	Remote Action RUN – SC ARRAY Create Directory on FileSystem single node
28	Remote Action RUN – SC ARRAY CCD Cooling single node
29	Remote Action RUN - SC ARRAY Get CCD Temperature single node
30	Remote Action RUN - SC ARRAY Camera Shot single node
31	Remote Action RUN - Telescope Goto
32	Remote Action RUN - Run External Script/Application
33	Remote Action RUN - SC ARRAY AutoFocus all node with LocalField method
34	Remote Action RUN - SC ARRAY AutoFocus single node with LocalField method
…	
500	VOYAGER General STATUS - Error (some error from action or thread raised)
501	VOYAGER General STATUS - Idle (nothing to do ready to work)
502	VOYAGER General STATUS - Action Running
503	VOYAGER General STATUS - Action Stopped
504	VOYAGER General STATUS - Undefined (just started Voyager … nothing defined)
505	VOYAGER General STATUS - Warning (some minor error from action or thread raised)
506	VOYAGER General STATUS - Unknow (Internal Automa cannot understand what asked to Voyager)

Example:


```
{"Event":"Signal","Timestamp":1550018150.45152,"Host":"hal9000","Inst":1,"Code":18}
```


d) NewFITReady

New FIT file just saved from Voyager to the O.S. filesystem.

Attribute	Type	Description
`File`	String	Path and name with extension of the file saved (usually referred to the server local

		disc if start with a drive letter unit, or to a network sharing if start with \\. Remember that \ is a special escape char and must be associate with an \ before. For network sharing be sure to have permission to read file
`Type`	Integer	the number represent the kind of image. See table below.

Type	Description
0	LIGHT
1	BIAS
2	DARK
3	FLAT

Example:


```
{"Event":"NewFITReady","Timestamp":1550018163.09996,"Host":"hal9000","Inst":1,"File":"C:\\Users\\leonardo\\Documents\\Voyager\\FIT\\TestShot_20190213_003550.fit","Type":0}
```


e) Shutdown

Voyager Application Server will be closed due to users request of application closing (user click on Voyager close button) or process was killed by O.S. . You must close client because connection is not available.

Example:


```
{"Event":"ShutDown","Timestamp":1548806904.00159,"Host":"hal9000","Inst":1}
```


f) RemoteActionResult

A remote action was ended in the server. You could check if you have task waiting for it matching the UID inside the event. Usually all the actions callable have this event at finish running except some services commands. Result of action is inside the event.

Attribute	Type	Description
`UID`	String	This is a unique string that identify in univocal way the action that have generated this result. It's a GUID String that was created automatically when you have created a new action command.
`ActionResultInt`	Integer	Result code of Action. See table below.
`Motivo`	String	If the `ActionResultInt` correspond to error in this field you'll find the description of the error.
`ParamRet`	Array	If the action related return parameters you'll found in this Array. Reference to each command to know which are the possible parameters.

ActionResultInt	Description	Note
0	NEED INIT	Wait to Running
1	READY	Ready to Running
2	RUNNING	Running

3	PAUSE	Paused
4	**OK**	Finished
5	FINISHED ERROR	Finished with Error
6	ABORTING	Abort request waiting during running
7	ABORTED	Finished aborted
8	TIMEOUT	Finished timeout
9	TIME END	Finished for timer end
10	**OK PARTIAL**	FInished with some task not executed

Example:

{"Event":"RemoteActionResult","Timestamp":1556621977.1658,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{"DownloadAndSaveTime":3.0700113}}

g) ArrayElementData

Contains data about status and controls from remote server. Usually used in Telescope Array management can be used to know status of a single server. Events arrive after a `GetArrayElementData` command.

Attribute	Type	Description
ROTCONN	Boolean	Indicate if rotator is connected or not. True = connected. False if not connected or control is empty
PAROT	Number	PA of rotator. Value of 1000 mean ND (not defined data) else value is expressed in degree
ROTROT	Boolean	Indicate if rotator is rotating or not
CCDCONN	Boolean	Indicate if camera is connected or not. true = connected. false if not connected or control is empty
CCDTEMP	Number	Temperature of camera peltier. Value of 1000 mean ND (not defined data) else value is expreseds in °Celsius
CCDPOW	Number	Power % of camera peltier. Value of 1000 mean ND (not defined data) else value is expressed in %
FOCCONN	Boolean	Indicate if focuser is connected or not. true = connected. false if not connected or control is empty
FOCPOS	Number	Step position of focuser. Value of -1000000 mean ND (not defined data) else value is expressed in step
FOCMOV	Boolean	Indicate if focuser is moving or not
FOCTEMP	Number	Temperature of focuser sensor. Value of -1000000 mean ND (not defined data) else value is expressed in °Celsius or ADU units (depends on focuser driver)
FOCHFD	Number	HFD value obtained in the last autofocus action (local or remote). Value of -1000 mean ND (not defined data) else value is expressed in pixel

Example:

{"Event":"ArrayElementData","Timestamp":1556117138.91959,"Host":"hal9000","Inst":1,"ROTCONN":false,"PAROT":1000,"ROTROT":false,"CCDCONN":false,"CCDTEMP":1000,"CCDPOW":1000,"FOCCONN":false,"FOCPOS":-1000000,"FOCMOV":false,"FOCTEMP":-1000000,"FOCHFD":-1000}

5. Commands

VOYAGER provides an RPC (remote procedure call) interface for clients. The message protocol is JSON RPC 2.0.

Requests are sent as a single line of text, terminated by CR LF. Responses from the server are also a single line of text terminated by CR LF. Pamaters name and parameters value are case sensitive, please for Boolean value use ***true*** or ***false*** lower case.

All the commands (exceptions you'll find in a single command description) return an **async** jsonrpc result or jsonrpc error. You can refer to jsonrpc protocol or see the example below. Remember that ID is a integer counter sequential of the command in the client scope.

All the commands (exceptions you'll find in a single command description) return **when finished** an `RemoteActionResult` event.

All Command (exceptions you'll find in a single command description) have like params a string unique identifier UID, usually used is a windows guide identifier https://en.wikipedia.org/wiki/Universally_unique_identifier . You can use anyway a unique string generated with your rule. This string must identify univocue the command.

Some commands can generate dedicated signal events before to send the `RemoteActionResult` final event.

Here is an example exchange between client (➔) and server (🡐):

Remote Setup Connect :

➔{"method": "RemoteSetupConnect", "params": {"UID":"69e329c8-c80d-416e-94f5-5862399446b6","TimeoutConnect":90}, "id": 22}

🡐{"jsonrpc": "2.0", "result": 0, "id": 22}

🡐{"Event":"Signal","Timestamp":1556983812.21223,"Host":"hal9000","Inst":1,"Code":15}

🡐{"Event":"RemoteActionResult","Timestamp":1556983826.98443,"Host":"hal9000","Inst":1,"UID":"69e329c8-c80d-416e-94f5-5862399446b6","ActionResultInt":4,"Motivo":"","ParamRet":{}}

Remote Setup Connect (error):

➔{"method": "RemoteSetupConnect", "params": {"UID":"32806c14-5820-4291-979a-71ba62004d96","TimeoutConnect":90}, "id": 3}

🡐{"jsonrpc": "2.0", "error": {"code": 1, "message": "could not connect all controls : Camera Error"}, "id": 3}

Remote Camera Shot :

➔ {"method": "RemoteCameraShot", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","Expo":10,"Bin":1,"IsROI":false,"ROITYPE":0,"ROIX":0,"ROIY":0,"ROIDX":0,"ROIDY":0,"FilterIndex":0,"ExpoType":0,"SpeedIndex":0,"ReadoutIndex":0,"IsSaveFile":true,"FitFileName":"%%fitdir%%\\TestShot_20190130_001330.fit"}, "id": 306}

🡐{"Event":"Signal","Timestamp":1556621998.29079,"Host":"hal9000","Inst":1,"Code":18}

←{"Event":"NewFITReady","Timestamp":1556622011.27632,"Host":"hal9000","Inst":1,"File":"C:\\Users\\leonardo\\Documents\\Voyager\\FIT\\TestShot_20190130_001330.fit","Type":0}

← {"Event":"Signal","Timestamp":1556622011.29079,"Host":"hal9000","Inst":1,"Code":2}

←{"Event":"RemoteActionResult","Timestamp":1556622011.30635,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{"DownloadAndSaveTime":3.0471478}}

Remote Setup Disconnect :

→{"method": "RemoteSetupDisconnect", "params": {"UID":"d4522a50-bf00-4bdd-acaa-19082578b9a0","TimeoutDisconnect":90}, "id": 9384}

←{"jsonrpc": "2.0", "result": 0, "id": 9384}

←{"Event":"Signal","Timestamp":1556989070.50118,"Host":"hal9000","Inst":1,"Code":16}

←{"Event":"RemoteActionResult","Timestamp":1556989071.28799,"Host":"hal9000","Inst":1,"UID":"d4522a50-bf00-4bdd-acaa-19082578b9a0","ActionResultInt":4,"Motivo":"","ParamRet":{}}

→{"method": "disconnect", "id": 1}

←{"jsonrpc": "2.0", "result": 0, "id": 1}

Close Your Client :

→{"method": "disconnect", "id": 1}

←{"jsonrpc": "2.0", "result": 0, "id": 1}

a) Disconnect

Method	`disconnect`
Description	Disconnect the Client from the Server. Necessary when you want to close the communication with server in a clean way. Just closing the socket without disconnect command force the server to wait heartbeat timeout to declare closed the communication and release the client thread. Using this command close immediately the connection and the thread. No `RemoteActionResult` will be received about this command
Params	None
Result	Integer(0)
License Required	*Base, Advanced, Full, Custom*

→{"method": "disconnect", "id": 1}

←{"jsonrpc": "2.0", "result": 0, "id": 1}

b) GetArrayElementData

Method	`GetArrayElementData`
Description	Ask to the Server to send the common data for Array Custom Management System . Status, CCD temperature, Rotator PA, Mount position, etc.etc. Data arrive like event. See the relative event `ArrayElementData`
Params	None
Result	Integer(0)
License Required	*Base, Advanced, Full, Custom*

➔{"method": "GetArrayElementData", "id": 6}

🡐{"jsonrpc": "2.0", "result": 0, "id": 6}

🡐{"Event":"ArrayElementData","Timestamp":1556117138.91959,"Host":"hal9000","Inst":1,"ROTCONN":false,"PAROT":1000,"ROTROT":false,"CCDCONN":false,"CCDTEMP":1000,"CCDPOW":1000,"FOCCONN":false,"FOCPOS":-1000000,"FOCMOV":false,"FOCTEMP":-1000000,"FOCHFD":-1000}

c) RemoteActionAbort

Method	`RemoteActionAbort`
Description	Ask to the Server to abort the action running
Params	UID (String): Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
Result	Integer(0)
License Required	*Base, Advanced, Full, Custom*
Remote Action Result Parameters	DownloadAndSaveTime (Number): Present only if Action is RemoteCameraShot , time remaing to finish the exposure in negative if action was aborted

➔{"method": "RemoteActionAbort", "params": {"UID":"e3f31937-8cac-4ac4-aad8-a0940f9cb2d4"}, "id": 127}

🡐{"jsonrpc": "2.0", "result": 0, "id": 127}

🡐{"Event":"Signal","Timestamp":1556719941.54408,"Host":"hal9000","Inst":1,"Code":2}

🡐{"Event":"RemoteActionResult","Timestamp":1556719941.58675,"Host":"hal9000","Inst":1,"UID":"e3f31937-8cac-4ac4-aad8-a0940f9cb2d4","ActionResultInt":7,"Motivo":"","ParamRet":{"DownloadAndSaveTime":-97.8279968}}

🡐{"Event":"Signal","Timestamp":1556719941.69196,"Host":"hal9000","Inst":1,"Code":505}

d) RemoteCameraShot

<table>
<tr><td>Method</td><td>RemoteCameraShot</td></tr>
<tr><td>Description</td><td>Ask to the Server to do an exposure with the parameters send. This method is ASync , a JSonRPC result will be send from server immediately with the answer to command. A RemoteActionResult event with the final result of the remote action will be send. Referring to the original command will be done with the UID. This mean in RemoteActionResult you find in the UID the same that used in the command call. Setup must be connected to get a shot. Also a NewFITReady event will be send to client if a remote shot was finished and file saved on disk. Sequence of command is send command, receive JSonRPC result, receive NewFITReady when shot is finished, receive and RemoteActionResult whit command final result.</td></tr>
<tr><td>Params</td><td>
<table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort</td></tr>
<tr><td>Expo</td><td>Number</td><td>Time of exposure expressed in seconds</td></tr>
<tr><td>Bin</td><td>Integer</td><td>Binning value for x and y</td></tr>
<tr><td>IsROI</td><td>Boolean</td><td>true if you want to use some kind of ROI, false for full framing</td></tr>
<tr><td>ROITYPE</td><td>Integer</td><td>See table below</td></tr>
<tr><td>ROIX</td><td>Integer</td><td>ROI x origin in pixel</td></tr>
<tr><td>ROIY</td><td>Integer</td><td>ROI y origin in pixel</td></tr>
<tr><td>ROIDX</td><td>Integer</td><td>ROI width x in pixel</td></tr>
<tr><td>ROIDY</td><td>Integer</td><td>ROI width y in pixel</td></tr>
<tr><td>FilterIndex</td><td>Integer</td><td>Index of filter to user for exposure like received in RemoteGetFiltersConfiguration or 0 for DSLR or COLOR CCD or no filter camera setup</td></tr>
<tr><td>ExpoType</td><td>Integer</td><td>See table of types in NewFITReady event</td></tr>
<tr><td>SpeedIndex</td><td>Integer</td><td>Index of filter to user for exposure like received in RemoteGetSpeedConfiguration or 0 for default</td></tr>
<tr><td>ReadoutIndex</td><td>Integer</td><td>Index of filter to user for exposure like received in RemoteGetReadoutConfiguration or 0 for default</td></tr>
<tr><td>IsSaveFile</td><td>Boolean</td><td>true always</td></tr>
<tr><td>FitFileName</td><td>String</td><td>Name of File to save , You must use \ for escape char like \ . You can use a special symbols to identify the location where to save file in the directory default of server, use %%fitdir%% to save FIT File in the default directory used by Voyager for general porpoise FIT. Use %%sequencedir%% for save file in the directory used by Voyager to save Sequence file.</td></tr>
</table>
</td></tr>
<tr><td>Result</td><td>Integer(0)</td></tr>
<tr><td>License Required</td><td>Custom</td></tr>
<tr><td>Remote Action Result Parameters</td><td>
<table>
<tr><td>DownloadAndSaveTime</td><td>Number</td><td>Time necessary for download data from camera</td></tr>
</table>
</td></tr>
</table>

ROITYPE	Description
-1	Custom ROI, you can define all ROI start and size parameters (ROIX,ROIY,ROIDX,ROIDY)
0	FullFrame ROI , ROI start and size parameters are ignored
1	Half Frame ROI , ROI start and size parameters are ignored
2	Quarter Frame ROI , ROI start and size parameters are ignored
3	1/8 Frame ROI , ROI start and size parameters are ignored
4	1/16 Frame ROI , ROI start and size parameters are ignored
5	Custom size Centered ROI, ROIX and ROIY parameter will be ignored ROIDX and ROYDY will be used

➔{"method": "RemoteCameraShot", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","Expo":10,"Bin":1,"IsROI":false,"ROITYPE":0,"ROIX":0,"ROIY":0,"ROIDX":0,"ROIDY":0,"FilterIndex":0,"ExpoType":0,"SpeedIndex":0,"ReadoutIndex":0,"IsSaveFile":true,"FitFileName":"%%fitdir%%\\TestShot_20190130_001330.fit"}, "id": 306}

➔{"Event":"Signal","Timestamp":1556621998.29079,"Host":"hal9000","Inst":1,"Code":18}

➔{"Event":"NewFITReady","Timestamp":1556622011.27632,"Host":"hal9000","Inst":1,"File":"C:\\Users\\leonardo\\Documents\\Voyager\\FIT\\TestShot_20190130_001330.fit","Type":0}

➔ {"Event":"Signal","Timestamp":1556622011.29079,"Host":"hal9000","Inst":1,"Code":2}

➔{"Event":"RemoteActionResult","Timestamp":1556622011.30635,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{"DownloadAndSaveTime":3.0471478}}

e) RemoteCooling

Method	`RemoteCooling`		
Description	Activate or Deactivate Camera Cooling . It's possible to do SetPoint, cooling down, warmup. Sync or ASync		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	IsSetPoint	Boolean	true for Cooling camera using internal firmware ramp
	IsCoolDown	Boolean	true for Cooling camera using Voyager ramp like configured in server
	IsASync	Boolean	If true action finish when cooling or warmup action is finished
	IsWarmup	Boolean	true for Warmup camera according ramp of warmup configured in Voyager server
	IsCoolerOFF	Boolean	true for Switch off cooling of camera
	Temperature	Number	Temperature to reach in cooling
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

➔{"method": "RemoteCooling", "params": {"UID":"37f4962a-73c5-44f5-80e1-d29f029f49a9","IsSetPoint":true,"IsCoolDown":false,"IsASync":true,"IsWarmup":false,"IsCoolerOFF":false,"Temperature":-25}, "id": 84}

➔{"jsonrpc": "2.0", "result": 0, "id": 84}

➔{"Event":"Signal","Timestamp":1556728960.12891,"Host":"hal9000","Inst":1,"Code":19}

➔{"Event":"Signal","Timestamp":1556728960.17578,"Host":"hal9000","Inst":1,"Code":2}

➔{"Event":"RemoteActionResult","Timestamp":1556728960.20703,"Host":"hal9000","Inst":1,"UID":"37f4962a-73c5-44f5-80e1-d29f029f49a9","ActionResultInt":4,"Motivo":"","ParamRet":{}}

f) RemoteCreateDir

Method	`RemoteCreateDir`		
Description	Create a directory in the remote Voyager server PC		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	Dir	String	Full Path and name of directory to create , You must use \ for escape char like \ or “. You can use a special symbols to identify the location where to create the directory , use %%fitdir%% to create Directory inside the default directory used by Voyager for general pourpose FIT. Use %%sequencedir%% for create the directory inside he directory used by Voyager to save Sequence file.
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

➔{"method": "RemoteCreateDir", "params": {"UID":"62967a0f-3076-4b53-bfe2-028b37407075","Dir":"%%sequencedir%%\\M12\\2019-05-01"}, "id": 1544}

➔{"jsonrpc": "2.0", "result": 0, "id": 1544}

➔{"Event":"Signal","Timestamp":1556734985.077,"Host":"hal9000","Inst":1,"Code":27}

➔{"Event":"RemoteActionResult","Timestamp":1556734985.21763,"Host":"hal9000","Inst":1,"UID":"62967a0f-3076-4b53-bfe2-028b37407075","ActionResultInt":4,"Motivo":"","ParamRet":{}}

g) RemoteFilterChangeTo

Method	`RemoteFilterChangeTo`		
Description	Change actual filter in the filter wheel		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide

			Window identifier or a unique key string generated
	FilterIndex	Integer	Index of filter to user for exposure like received in `RemoteGetFiltersConfiguration` or 0 for DSLR or COLOR CCD or no filter camera setup
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

➔{"method": "RemoteFilterChangeTo", "params": {"UID":"82f79427-d192-4b09-81ed-0d363d96d6de","FilterIndex":2}, "id": 2607}

➔{"jsonrpc": "2.0", "result": 0, "id": 2606}

➔{"Event":"Signal","Timestamp":1556735516.84362,"Host":"hal9000","Inst":1,"Code":24}

➔{"Event":"RemoteActionResult","Timestamp":1556735521.89267,"Host":"hal9000","Inst":1,"UID":"82f79427-d192-4b09-81ed-0d363d96d6de","ActionResultInt":4,"Motivo":"","ParamRet":{}}

h) RemoteFilterGetActual

Method	`RemoteFilterGetActual`		
Description	Get index of actual filter in the filter wheel		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		
Remote Action Result Parameters	FilterIndex	Integer	Index of filter to user for exposure like received in `RemoteGetFiltersConfiguration` or -1 if there's not filter wheel or filter to get.

➔{"method": "RemoteFilterGetActual", "params": {"UID":"ffc14de0-feee-4417-bb28-c4410c2c1d0d"}, "id": 3762}

➔{"jsonrpc": "2.0", "result": 0, "id": 3762}

➔{"Event":"RemoteActionResult","Timestamp":1556736091.15078,"Host":"hal9000","Inst":1,"UID":"ffc14de0-feee-4417-bb28-c4410c2c1d0d","ActionResultInt":4,"Motivo":"","ParamRet":{"FilterIndex":2}}

i) RemoteFlat

Method	`RemoteFlat`		
Description	Execute Flat Sequence in Remote Voyager Server		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	IsOnlyForRemote	Boolean	Use always true
	RemoteConfigurationFile	String	Only File name with extension of Voyager Sequence Flat File to use
	DataBase64	String	File data of the Sequence Flat File to use converted to Base64 coding ascii text
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

➔{"method": "RemoteFlat", "params": {"UID":"3a7a6e74-5a67-4471-b0c5-1e7199bff755","IsOnlyForRemote":true,"RemoteConfigurationFile":"test.s2f","DataBase64":" pFbnZlbG ……. *[Missing a lot of data]* ……. 9wZT4NCg=="}, "id": 161}

🡐{"jsonrpc": "2.0", "result": 0, "id": 160}

🡐{"Event":"Signal","Timestamp":1556790000.43286,"Host":"hal9000","Inst":1,"Code":23}

🡐{"Event":"RemoteActionResult","Timestamp":1556790014.36533,"Host":"hal9000","Inst":1,"UID":"3a7a6e74-5a67-4471-b0c5-1e7199bff755","ActionResultInt":4,"Motivo":"","ParamRet":{}}

j) RemoteFocus

Method	`RemoteFocus`		
Description	Execute AutoFocus Action in Remote Voyager Server		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	IsRoboFireLocalField	Boolean	true if you want to use the RoboFire LocalField Autofocus on all CCD Frame, false to use RoboFire with RoboStar selection on single star
	IsAsyncMode	Boolean	Always use true
	filtroFuocoIndex	Integer	Index of filter to user for focuse like received in `RemoteGetFiltersConfiguration` or 0 for DSLR or COLOR CCD or no filter camera setup
	IsWDMaxHFDVariation	Boolean	true if you want repeat focus if result HFD is greater than a certain variation value in percentuage in the last autofocus HFD mobile mean
	WDMaxHFDLimitVariation	Number	Max value percentuage of HFD variation

			considered good
	IsWDMaxHFD	Boolean	true if you want repeat focus if result HFD is great than a certain value in pixel
	WDMaxHFDLimit	Number	Max value in pixel of final HFD considered good
	IsRetryFocusOnWD	Boolean	true to retry autofocus if ones of the WD is happen or false to return to previous focus position
	PreviousPosition	Integer	Value in step of previous focus position
	IsFMAcquireStarFocus	Boolean	true if you want to use FocusMax Acquire Star routine (use false to use Voyager RoboStar)
	StarName	String	Name of focus star If you want to use a defined star for Focus on Star mode
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

➔{"method": "RemoteFocus", "params": {"UID":"dd486bd0-b141-43e8-a401-4871cea992f4","IsRoboFireLocalField":false,"IsAsyncMode":true,"filtroFuocoIndex":1,"IsWDMaxHFDVariation":false,"WDMaxHFDLimitVariation":0,"IsWDMaxHFD":false,"WDMaxHFDLimit":9.4,"IsRetryFocusOnWD":true,"PreviousPosition":-1,"IsFMAcquireStarFocus":false,"StarName":""}, "id": 1792}

➔{"jsonrpc": "2.0", "result": 0, "id": 1792}

➔{"Event":"Signal","Timestamp":1556790810.28741,"Host":"hal9000","Inst":1,"Code":5}

➔{"Event":"RemoteActionResult","Timestamp":1556790835.42092,"Host":"hal9000","Inst":1,"UID":"dd486bd0-b141-43e8-a401-4871cea992f4","ActionResultInt":5,"Motivo":"Focus Async Error (Error executing VCurve AutoFocus : Maxim iteration to find focus side HFD reached)","ParamRet":{}}

k) RemoteFocuserMoveTo

Method	`RemoteFocuserMoveTo`		
Description	Move the focuser to the position asked in Remote Voyager Server		
Params			
	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	IsAbsoluteMove	Boolean	true if you want to move to absolute position, false to move by offset relative to actual position
	NewPosition	Integer	Position in step (or offset)
	MoveDirection	Integer	Direction where to move in case of offset, see table below. Zero for Absolute movements.

	IsBLCompensation	Boolean	true if you want apply a backlash compensation to movements
	BLCompVersus	Integer	Versus of compensation, see table below. Zero if you don't use compensation
	BLCompStep	Integer	Compensation steps to apply
	IsFinalPositionCheck	Boolean	true if you want check final position of focuser when the driver return command finished. Some focuser driver can return command finished but focuser motor not yet finished. If Voyager found different position from what asked retry the command
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

MoveDirection	Description
0	OUT
1	IN

➔{"method": "RemoteFocuserMoveTo", "params": {"UID":"84a92e1e-7383-4854-9c36-dbc77351836f","IsAbsoluteMove":true,"NewPosition":5000,"MoveDirection":0,"IsBLCompensation":true,"BLCompVersus":1,"BLCompStep":0,"IsFinalPositionCheck":true}, "id": 72}

➔{"jsonrpc": "2.0", "result": 0, "id": 72}

➔{"Event":"Signal","Timestamp":1556983836.33518,"Host":"hal9000","Inst":1,"Code":20}

➔{"Event":"RemoteActionResult","Timestamp":1556983849.47281,"Host":"hal9000","Inst":1,"UID":"84a92e1e-7383-4854-9c36-dbc77351836f","ActionResultInt":4,"Motivo":"","ParamRet":{}}

l) RemoteFocuserOffset

Method	`RemoteFocuserOffset`		
Description	Move the focuser relative from actual position by offset in Remote Voyager Server		
Params			
	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	Offset	Integer	Offset in step, use positive or negative value
	IsBLCompensation	Boolean	true if you want apply a backlash compensation to movements
	BLCompVersus	Integer	Versus of compensation, see table below. Zero if you don't use compensation
	BLCompStep	Integer	Compensation steps to apply

	IsFinalPositionCheck	Boolean	true if you want check final position of focuser when the driver return command finished. Some focuser driver can return command finished but focuser motor not yet finished. If Voyager found different position from what asked retry the command
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

MoveDirection	Description
0	OUT
1	IN

➔{"method": "RemoteFocuserOffset", "params": {"UID":"84a92e1e-7383-4854-9c36-dbc77351836f", "Offset":-200,"IsBLCompensation":true,"BLCompVersus":1,"BLCompStep":0,"IsFinalPositionCheck":true}, "id": 73}

➔{"jsonrpc": "2.0", "result": 0, "id": 73}

➔{"Event":"Signal","Timestamp":1556983836.33518,"Host":"hal9000","Inst":1,"Code":21}

➔{"Event":"RemoteActionResult","Timestamp":1556983849.47281,"Host":"hal9000","Inst":1,"UID":"84a92e1e-7383-4854-9c36-dbc77351836f","ActionResultInt":4,"Motivo":"","ParamRet":{}}

m) RemoteGetCCDTemperature

Method	`RemoteGetCCDTemperature`		
Description	Return temperature of CCD Chamber from Remote Voyager Server		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		
Remote Action Result Parameters	CCDTemp	Number	Temperature °C or ADU Value

➔{"method": "RemoteGetCCDTemperature", "params": {"UID":"24a92e1e-7383-4854-9c36-dbc77351836f"}, "id": 173}

🡐{"jsonrpc": "2.0", "result": 0, "id": 173}

🡐{"Event":"RemoteActionResult","Timestamp":1556985994.19153,"Host":"hal9000","Inst":1,"UID":"24a92e1e-7383-4854-9c36-dbc77351836f","ActionResultInt":4,"Motivo":"","ParamRet":{"CCDTemp":10}}

n) RemoteGetFilterConfiguration

Method	RemoteGetFilterConfiguration		
Description	Return data about filters configuration from Remote Voyager Server. ATTENTION! Filter returned in this command are listed base 1, filter index used in other commands are base 0. Filter1 here is index 0 in other commands, Filter2 here is index 1 in other command, … and so		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		
Remote Action Result Parameters	FilterNum	Integer	Number of filters in remote Filter Wheels
	Filter1_Name	String	Name of filter 1
	Filter1_MagMin	Number	Min Magnitude of stars for focus, filter 1
	Filter1_MagMax	Number	Max Magnitude of stars for focus, filter 1
	Filter1_Offset	Integer	Offset in step for focus relative to this filter, filter 1
	..repeat for FilterNum times for each filter		

➔{"method": "RemoteGetFilterConfiguration", "params": {"UID":"cc7b1c6d-48a6-418f-a02b-2e8f1ece1750"}, "id": 4840}

🡐{"jsonrpc": "2.0", "result": 0, "id": 4840}

🡐{"Event":"RemoteActionResult","Timestamp":1556986227.4567,"Host":"hal9000","Inst":1,"UID":"cc7b1c6d-48a6-418f-a02b-2e8f1ece1750" ,"ActionResultInt":4, "Motivo":"", "ParamRet" :{"FilterNum":5, "Filter1_Name":"L","Filter1_MagMin":4,"Filter1_MagMax":7,"Filter1_Offset":0,"Filter2_Name":"R","Filter2_MagMin":4,"Filter2_MagMax":7,"Filter2_Offset":0,"Filter3_Name":"G","Filter3_MagMin":4,"Filter3_MagMax":7,"Filter3_Offset":0,"Filter4_Name":"B","Filter4_MagMin":4,"Filter4_MagMax":7,"Filter4_Offset":0,"Filter5_Name":"HA","Filter5_MagMin":4,"Filter5_MagMax":7,"Filter5_Offset":0}}

o) RemoteGetReadoutConfiguration

Method	`RemoteGetReadoutConfiguration`
Description	Return data about CCD Readout Mode configuration from Remote Voyager Server
Params	UID — String — Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
Result	Integer(0)
License Required	*Base, Advanced, Full, Custom*
Remote Action Result Parameters	ReadoutNum — Integer — Number of Readout Mode in remote CCD Readout1_Name — String — Name of Readout Mode 1 Readout1_Index — Number — Index of Readout Mode 1 ..repeat for ReadoutNum times for each readout mode

➔{"method": "RemoteGetReadoutConfiguration", "params": {"UID":"94ac2036-0e2e-49f4-a56b-268fd43d3072"}, "id": 7304}

➔{"jsonrpc": "2.0", "result": 0, "id": 7304}

➔{"Event":"RemoteActionResult","Timestamp":1556987465.42752,"Host":"hal9000","Inst":1,"UID":"94ac2036-0e2e-49f4-a56b-268fd43d3072","ActionResultInt":4 ,"Motivo":"","ParamRet": {"ReadoutNum":1,"Readout1_Name":"Default","Readout1_Index":0}}

p) RemoteGetSpeedConfiguration

Method	`RemoteGetSpeedConfiguration`
Description	Return data about CCD Speed Mode configuration from Remote Voyager Server
Params	UID — String — Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
Result	Integer(0)
License Required	*Base, Advanced, Full, Custom*
Remote Action Result Parameters	SpeedNum — Integer — Number of Readout Mode in remote CCD Speed1_Name — String — Name of Speed Mode 1 Speed1_Index — Number — Index of Speed Mode 1 ..repeat for SpeedNum times for each speed mode

➔{"method": "RemoteGetSpeedConfiguration", "params": {"UID":"c012d391-3a7a-4cc3-9dc6-9593e4812d36"}, "id": 7904}

➔{"jsonrpc": "2.0", "result": 0, "id": 7904}

➔{"Event":"RemoteActionResult","Timestamp":1556988329.07105,"Host":"hal9000","Inst":1,"UID":"c012d391-3a7a-4cc3-9dc6-9593e4812d36","ActionResultInt":4,"Motivo":"", "ParamRet": {"SpeedNum":5,"Speed1_Name":"ISO 100","Speed1_Index":0,"Speed2_Name":"ISO 200","Speed2_Index":1,"Speed3_Name":"ISO 400","Speed3_Index":2,"Speed4_Name":"ISO 800","Speed4_Index":3,"Speed5_Name":"ISO 1600","Speed5_Index":4}}

q) RemoteRotatorMoveTo

Method	`RemoteRotatorMoveTo`		
Description	Move the rotator to the PA requested in Remote Voyager Server		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	PA	Number	Position angle in Degree
	IsWaitAfter	Boolean	true if you want to wait an interval of seconds after driver report rotation finished
	WaitAfterSeconds	Integer	Number of second to wait
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

➔{"method": "RemoteRotatorMoveTo", "params": {"UID":"a53c6e8a-be1d-4c67-8ed7-df41c15d8923","PA":0,"IsWaitAfter":false,"WaitAfterSeconds":5}, "id": 9423}

➔{"jsonrpc": "2.0", "result": 0, "id": 9423}

➔{"Event":"Signal","Timestamp":1556989105.71688,"Host":"hal9000","Inst":1,"Code":22}

➔{"Event":"RemoteActionResult","Timestamp":1556989126.85292,"Host":"hal9000","Inst":1,"UID":"a53c6e8a-be1d-4c67-8ed7-df41c15d8923","ActionResultInt":4,"Motivo":"","ParamRet":{}}

r) RemoteRunExternal

Method	`RemoteRunExternal`		
Description	Execute a script or an executable in Remote Voyager Server		
Params	UID	String	Unique identifier of the Action to abort. Use

			a Guide Window identifier or a unique key string generated
	FileName	String	Full Path and script name file with extension. \\ instead to \ for escape chars.
	Arguments	String	Arguments to pass in command line when calling script or executable
	TimeoutMilliseconds	Integer	Number of seconds to wait finish of running
	WaitFinish	Boolean	true if you want to wait finish of execute
	TryKillOnTimeout	Boolean	true if at wait finished for timeout Voyager must try to kill the process running
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

➔{"method": "RemoteRunExternal", "params": {"UID":"a53c6e8a-be1d-4c67-8ed7-df41c15d8923","FileName":"notepad.exe","Arguments":"pippo.txt","TimeoutMilliseconds":10000,"WaitFinish":false,"TryKillOnTimeout":false}, "id": 19423}

➔{"jsonrpc": "2.0", "result": 0, "id": 19423}

➔{"Event":"Signal","Timestamp":1556990521.19391,"Host":"hal9000","Inst":1,"Code":32}

➔{"Event":"RemoteActionResult","Timestamp":1556990521.31099,"Host":"hal9000","Inst":1,"UID":"a53c6e8a-be1d-4c67-8ed7-df41c15d8923","ActionResultInt":4,"Motivo":"","ParamRet":{}}

s) RemoteSetupConnect

Method	`RemoteSetupConnect`		
Description	Connect all controls Setup in Remote Voyager Server. You can also send command if all controls is already connect or you can send also you the previous time you ask connection but some controls result at command finish not connected. In the last case Voyager retry to connect only the control not connected.		
Params			
	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	TimeoutConnect	Integer	Number of seconds to wait before declaring connection timeout. Timeout happen also if time to wait is too short to allow all controls to connect to Voyager
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

➔{"method": "RemoteSetupConnect", "params": {"UID":"69e329c8-c80d-416e-94f5-5862399446b6","TimeoutConnect":90}, "id": 22}

←{"jsonrpc": "2.0", "result": 0, "id": 22}

←{"Event":"Signal","Timestamp":1556983812.21223,"Host":"hal9000","Inst":1,"Code":15}

←{"Event":"RemoteActionResult","Timestamp":1556983826.98443,"Host":"hal9000","Inst":1,"UID":"69e329c8-c80d-416e-94f5-5862399446b6","ActionResultInt":4,"Motivo":"","ParamRet":{}}

t) RemoteSetupDisconnect

Method	`RemoteSetupConnect`		
Description	Disconnect all controls Setup in Remote Voyager Server.		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	TimeoutDisconnect	Integer	Number of seconds to wait before declaring disconnection timeout. Timeout happen also if time to wait is too short to allow all controls to disconnect from Voyager
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

→{"method": "RemoteSetupDisconnect", "params": {"UID":"d4522a50-bf00-4bdd-acaa-19082578b9a0","TimeoutDisconnect":90}, "id": 9384}

←{"jsonrpc": "2.0", "result": 0, "id": 9384}

←{"Event":"Signal","Timestamp":1556989070.50118,"Host":"hal9000","Inst":1,"Code":16}

←{"Event":"RemoteActionResult","Timestamp":1556989071.28799,"Host":"hal9000","Inst":1,"UID":"d4522a50-bf00-4bdd-acaa-19082578b9a0","ActionResultInt":4,"Motivo":"","ParamRet":{}}

u) RemoteSolveActualPosition

Method	`RemoteSolveActualPosition`		
Description	Try to plate/blind solving actual position of telescope with a in Remote Voyager Server.		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	IsBlind	Boolean	true if you want to use Blind Solving engine, false for Plate Solving Engine

	IsSync | Boolean | true if you want to sync mount to the coordinates solved
Result	Integer(0)
License Required	*Base, Advanced, Full, Custom*
Remote Action Result Parameters	IsSolved | Boolean | True if solved LastError | String | Error if not solved RA | Number | RA in J2000 format where pointing telescope DEC | String | DEC in J2000 format where pointing telescope PA | Number | PA in Degree of camera

➔{"method": "RemoteSolveActualPosition", "params": {"UID":"d4522a50-bf00-4bdd-acaa-19082578b9a0","IsBlind":false,"IsSync":false}, "id": 9384}

🡐{"jsonrpc": "2.0", "result": 0, "id": 9384}

🡐{"Event":"NewFITReady","Timestamp":1557053647.49358,"Host":"hal9000","Inst":1,"File":"C:\\Users\\leonardo\\Documents\\Voyager\\FIT\\SyncVoyager_20190505_105358.fit","Type":0}

🡐{"Event":"Signal","Timestamp":1557053647.52483,"Host":"hal9000","Inst":1,"Code":25}

🡐{"Event":"Signal","Timestamp":1557053650.61527,"Host":"hal9000","Inst":1,"Code":2}

🡐{"Event":"RemoteActionResult","Timestamp":1557053650.64094,"Host":"hal9000","Inst":1,"UID":"d4522a50-bf00-4bdd-acaa-19082578b9a0","ActionResultInt":4 ,"Motivo":"","ParamRet":{"IsSolved":true,"LastError":"","RA":7.291651816591,"DEC":89.7363320162195,"PA":208.428127473733}}

v) RemoteSolveFITFile

Method	`RemoteSolveFITFile`
Description	Try to plate/blind solving a referenced FIT File with a in Remote Voyager Server.
Params	UID | String | Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated FileName | String | File FIT to solve with full path , replace \ with \\ IsBlind | Boolean | True if you want to use Blind Solving engine, False for Plate Solving Engine
Result	Integer(0)
License Required	*Base, Advanced, Full, Custom*

Remote Action Result Parameters	

IsSolved	Boolean	true if solved
LastError	String	Error if not solved
RA	Number	RA in J2000 format where pointing telescope
DEC	String	DEC in J2000 format where pointing telescope
PA	Number	PA in Degree of camera

➔{"method": "RemoteSolveFITFile", "params": {"UID":"d4522a50-bf00-4bdd-acaa-19082578b9a0","FileName":"C:\\Progetti\\Voyager2Release_2.0\\FIT\\M_65_LIGHT_L_600s_BIN1_-25C_001_20170415_220853_073_W.FIT","IsBlind":false }, "id": 9384}

➔{"jsonrpc": "2.0", "result": 0, "id": 9384}

➔{"Event":"RemoteActionResult","Timestamp":1557070480.10141,"Host":"hal9000","Inst":1,"UID":"d4522a50-bf00-4bdd-acaa-19082578b9a0","ActionResultInt":4,"Motivo":"","ParamRet":{"IsSolved":true,"LastError":"","RA":11.3153494744318,"DEC":13.0895540054556,"PA":359.255478270067}}

w) RemoteGetCCDSizeInfo

Method	`RemoteGetCCDSizeInfo`
Description	Return number of pixel in x y e dimension of pixel in microns from remote Voyager Server
Params	

UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated

Result	Integer(0)
License Required	*Base, Advanced, Full, Custom*
Remote Action Result Parameters	

DX	Integer	Number of pixels in X
DY	Integer	Number of pixels in Y
PixelSize	Number	Size of Pixel in microns

➔{"method": "RemoteGetCCDSizeInfo", "params": {"UID":"24a92e1e-7383-4854-9c36-dbc77351836f"}, "id": 173}

➔{"jsonrpc": "2.0", "result": 0, "id": 173}

←{"Event":"RemoteActionResult","Timestamp":1557075280.27633,"Host":"hal9000","Inst":1,"UID":"24a92e1e-7383-4854-9c36-dbc77351836f","ActionResultInt":4,"Motivo":"","ParamRet":{"DX":2048,"DY":2048,"PixelSize":7.4}}

6. Workflow

- Open connection to the server
- Read Socket Loop Start in a Thread
- You'll receive ones at beginning the `Version` Event FROM server
- You'll receive each 5s the `Polling` Event FROM server if no data ready from server to send
- Read and process the event received
- Send command if needed and wait response to command, reset your polling timer
- If you don't have nothing to send and polling timer passed the 5s send a polling event to avoid connection closing (don't stop to polling the server also during command result waiting)
- You'll receive `Shutdown` Event if Voyager will be closed during your connection
- When finished send `disconnect` command or close the socket.

Example of exchange with server from client connection to client close:

←{"Event":"Version","Timestamp":1550096193.55834,"Host":"hal9000","Inst":1,"VOYVersion":"Release 2.0.14f - Built 2019-02-11","VOYSubver":"","MsgVersion":1}

→{"Event":"Polling","Timestamp":1550096198.68338,"Host":"hal9000","Inst":1}

←{"Event":"Signal","Timestamp":1550096236.27807,"Host":"hal9000","Inst":1,"Code":18}

←{"Event":"Polling","Timestamp":1550096241.29392,"Host":"hal9000","Inst":1}

→{"Event":"Polling","Timestamp":1550096198.68338,"Host":"hal9000","Inst":1}

←{"Event":"NewFITReady","Timestamp":1550096247.10677,"Host":"hal9000","Inst":1,"File":"C:\\Users\\leonardo\\Documents\\Voyager\\FIT\\TestShot_20190213_221716.fit","Type":0}

←{"Event":"Signal","Timestamp":1550096247.13798,"Host":"hal9000","Inst":1,"Code":2}

→{"Event":"Polling","Timestamp":1550096252.1815,"Host":"hal9000","Inst":1}

→{"method": "disconnect", "id": 1}

←{"jsonrpc": "2.0", "result": 0, "id": 1}