STARKEEPER.IT

Voyager Application Server Protocol

Events, Methods and Workflow (TCP-IP)

Leonardo Orazi

December 27, 2024

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
    - environment manage

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. Authentication

Authentication level between server/client is defined in Voyager -> Setup -> Remote Tab. Possible is none (no authentication required), Username and Password (basic authentication needed with dedicated command from client), Ticket (for renting system, info only under NDA, contact Voyager support).

If the authorization level is not equal to NONE, server will wait for 5s after connection to receive the authentication request otherwise will close the connection). If the authentication fail the connection will be closed immediatly.

If the client is local and authorization is needed or the client will do authentication or the connection will be leave opened until the first command that need authentication will be asked and in this connection will be closed. Some commands and the events not need authentication and in this case a local client can run forever.

5. 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,"VOYVersion":"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.
`VoyType`	String	Logical FIT Type like managed in Voyager. See table below.
`SeqTarget`	String	Target Name if FIT was shot in a Sequence Running.

Type	Description
0	LIGHT
1	BIAS
2	DARK
3	FLAT

VoyType	Description
TEST	FIT saved by Voyager in a Simple Test Shot for general porpouse, not done during sequence
SHOT	FIT saved by Voyager during a Sequence or in a DragScript Exposure Block
SYNC	FIT saved by Voyager during Blind Solve or Plate Solve actions

Example:

{"Event":"NewFITReady","Timestamp":1550018163.09996,"Host":"hal9000","Inst":1,"File":"C:\\Users\\leonardo\\Documents\\Voyager\\FIT\\M81_20190213_003550.fit","Type":0,"VoyType":"SHOT","SeqTarget":"M81"}

e) NewJPGReady

If the client is in Dashboard mode a base64 data of the last FIT file stretched and compressed in JPG quality will be sended from the application server and plus a various info related. Sended only to Dashboard client.

Attribute	Type	Description
File	String	Path and name with extension of the file source of JPG stretch (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.
SequenceTarget	String	Target name if a Sequence is associated to this shot
TimeInfo	String	Time of file creation in local PC where running Voyager
TimeInfoUTC	String	Time of file creation in UTC
Expo	Numeric	Value of exposure time in seconds
Bin	Integer	Binning used for shot
Filter	String	Name of Filter used for shot
HFD	Numeric	HFD mean value of stars in shot
StarIndex	Numeric	Index of stars avalaible in the image related to understand eventually a cloud or veil in image. You must evaluate the trend of this value in various shot
PixelDimX	Integer	Larghezza in Pixel dell'immagine
PixelDimY	Integer	Altezza in Pixel dell'immagine
Base64Data	String	Data in base64 of a compressed jpg file ready to use in web img tag or to save like a jpg file

{"Event":"NewJPGReady","Timestamp":1564313171.92553,"Timestamp":1564311171.92553,"Host":"hal9000","Inst":1,"File":"C:\\Users\\leonardo\\Documents\\Voyager\\FIT\\TestShot_20190728_112558.fit","SequenceTarget":"","TimeInfo":1564313170.52465,"Expo":1,"Bin":2,"Filter":"** BayerMatrix **", "HFD":4.53,"StarIndex":8.21,
"PixelDimX":2048, "PixelDimY":1024,
"Base64Data":"/9j/4AAQSkZJRgABAQEAYABgAAD/……"}

f) 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}

g) 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}}
```


h) 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}

i) ControlData

Contains data about status and controls from remote server. Usually used for Dashboard and not avalaible for Telescope Array management can be used to know status of a controls in the connected server. Events arrive each 2s automatically if client is declared as Dashboard with `RemoteSetDashboardMode` command. Sended only to Dashboard client.

Attribute	Type	Description
TI	String	Actual Timing in Voyager format AAAA-MM-GG HH:MM:SS
TIUTC	String	UTC Timing in Voyager format AAAA-MM-GG HH:MM:SS
VOYSTAT	Integer	Actual Status of Voyager Application. See table below.
SETUPCONN	Boolean	Indicate if all setup controls in Voyager are connected with true or false
CCDCONN	Boolean	Indicate if camera control is connected or not. True = connected. False if not connected or control is empty
CCDTEMP	Number	Temperature of cooling in Camera. Some special value is possibile see table below.
CCDPOW	Number	Percentage of power used by Peltier
CCDSETP	Number	Temperature Set Point asked to Cooler
CCDCOOL	Boolean	True if Peltier is switched on or False if is switched off or Peltier is not present
CCDSTAT	Integer	Status of cooling automa inside Voyager
MNTCONN	Boolean	Indicate if mount control is connected or not. True = connected. False if not connected or control is empty
MNTPARK	Boolean	Indicate if mount is parked. True = connected. False if not connected or control is empty
MNTRA	String	Actual RA of Mount JNow
MNTDEC	String	Actual DEC of Mount JNow
MNTRAJ2000	String	Actual RA of Mount J2000

MNTDECJ2000	String	Actual DEC of Mount J2000
MNTAZ	String	Actual Azimuth of Mount
MNTALT	String	Actual Altitude of Mount
MNTPIER	String	Actual Pier of Mount (pierWest = Before Meridian, pierEast = After Meridian)
MNTTFLIP	String	Time to Meridian Cross in HH:mm:SS if negative mean is before
MNTSFLIP	Integer	Status of Meridian Flip in Voyager, See table below.
MNTTRACK	Boolean	Indicate if the mount is tracking
MNTSLEW	Boolean	Indicate if the mount is slewing
AFCONN	Boolean	Indicate if Autofocus is connected. True = connected. False if not connected or control is empty
AFTEMP	Numeric	Temperature coming from Focuser.Some special value is possibile see table below.
AFPOS	Numeric	Position of Focuser in Step. Some special value is possibile see table below.
SEQTOT	Integer	Total in seconds of all shot in a Sequence running
SEQPARZ	Integer	Total in seconds of elapse shot in a Sequence Running.
GUIDECONN	Boolean	Indicate if guide controls is connected or not. . True = connected. False if not connected or control is empty
GUIDESTAT	Integer	Status of guide inside Voyager
DITHSTAT	Integer	Status of Dithering inside Voyager
GUIDEX	Numeric	Guide error in pixels in X axis
GUIDEY	Numeric	Guide error in pixels in Y axis
PLACONN	Boolean	Indicate if planetarium controls is connected or not. . True = connected. False if not connected or control is empty
SEQNAME	String	Name of Sequence running
SEQSTART	String	hh:mm:ss of sequence start
SEQREMAIN	String	hh:mm:ss of remaining time to finish sequence
SEQEND	String	hh:mm:ss of sequence end
RUNSEQ	String	FileName of actually running Sequence, empty if no Sequence running
RUNDS	String	FileName of actually running DragScript, empty if no DragScript running
ROTCONN	Boolean	Indicate if rotator control is connected or not .. True = connected. False if not connected or control is empty
ROTPA	Numeric	Position Angle in Degree of the Rotator (-1 or ERROR VALUE mean unknow position)
ROTSKYPA	Numeric	Last Position Angle of the camera in the SKY like resolved in solving actions (-1 or ERROR VALUE = unknow position)
ROTISROT	Boolean	Indicate if the rotator is rotating. True = is rotating
DOMECONN	Boolean	Indicate if dome control is connected or not .. True = connected. False if not connected or control is empty
DOMEPA	Numeric	Position Angle in Degree of the Dome (-1 or ERROR VALUE mean unknow position)
DOMEISMOV	Boolean	Indicate if the dome is rotating or shutter is moving. True = is rotating / moving
DOMESHUTTER	String	Indicate the status of Shutter in ASCOM string rappresentation, see table below

VOYSTAT	Description	Note
0	STOPPED	Voyager is not connected with setup some actions cannot work
1	IDLE	Voyager can run action , actually is in idle
2	RUN	Voyager is running an action
3	ERRORE	Voyager is in idle but last action finished with error
4	UNDEFINED	Voyager status cannot be determined
5	WARNING	Voyager is in idle but last action finished with a warning

Special Values	Description	Note
-123456789	OFF VALUE	Control switched OFF or not present
+123456789	ERROR VALUE	Error in report value or control not present

CCDSTAT	Description	Note
0	INIT	Voyager application is initializing then Camera Control
1	UNDEF	Status not recognized
2	NO COOLER	No cooler for this camera
3	OFF	Cooler Off
4	COOLING	Cooling running
5	COOLED	Cooled
6	TIMEOUT COOLING	Timeout Cooling
7	WARMUP RUNNING	Warmup Running
8	WARMUP END	Warmup Finished
9	ERROR	Error in Camera Control

GUIDESTAT	Description	Note
0	STOPPED	
1	WAITING_SETTLE	Running but waiting to go under the limit max
2	RUNNING	
3	TIMEOUT_SETTLE	Running but cannot settled for timeout

DITHSTAT	Description	Note
0	STOPPED	
1	RUNNING	Running but waiting to go under the limit max
2	WAITING _SETTLE	
3	TIMEOUT_SETTLE	Running but cannot settled for timeout

DOMESHUTTER	Description	Note
0	shutterOpen	
1	shutterClosed	
2	shutterOpening	
3	shutterClosing	
4	shutterError	
5	ERROR	Internal error or unknow shutter status

MNTSFLIP	Description	Note
0	Not needed	Pier is West
1	To do	Meridian flip is necessary, Voyager waiting the right internally status
2	Running	In execution
3	Done	Pier is PierEast
4	Unmanageable	FORK Mount
5	ERROR	Internal error or unknow pier status

Example:

{"Event":"ControlData","Timestamp":1564675036.22405,"Host":"hal9000","Inst":1, "TI":"2019-08-02 19:24:32","SETUPCONN":true,"CCDCONN":true,"CCDTEMP":10,"CCDPOW":-123456789,"CCDSETP":123456789,"CCDCOOL":false,"CCDSTAT":1,"MNTCONN":true,"MNTPARK":false,"MNTRA":"02:49:50","MNTDEC":"47° 20' 07\"","MNTRAJ2000":"02:33:44","MNTDECJ2000":"47° 31' 17\"","MNTAZ":"331° 23' 32\"","MNTALT":"-16° 09' 55\"","MNTPIER":"pierEast","MNTTFLIP":"09:08:40", "MNTSFLIP":3, "MNTTRACK":true,"AFCONN":false,"AFTEMP":123456789,"AFPOS":123456789, "SEQTOT":0,"SEQPARZ":0,"GUIDECONN":true,"GUIDESTAT":2,"DITHSTAT":0,"GUIDEX":-0.259,"GUIDEY":0.039,"PLACONN":false ……. }

j) WeatherAndSafetyMonitorData

Contains connection status and data about Weather System Control and Safety Monitor Controls from remote server. Usually used for Dashboard and not avalaible for Telescope Array. Events arrive each 30s automatically if client is declared as Dashboard with `RemoteSetDashboardMode` command. Sended only to Dashboard client.

Attribute	Type	Description
`WSConnected`	boolean	True if weather control is configured in Voyager and Data Read Process work correctly
`SMConnected`	boolean	True if Safety Monitor control is configured in Voyager and connected
`SMStatus`	string	String of Safety Monitor Control Status SAFE or UNSAFE or empty string
`WSCloud`	string	Cloud status of Weather control (UNKNOW,CLEAR,CLOUDY,VERY_CLOUDY)
`WSRain`	string	Rain status of Weather control (UNKNOW,DRY,WET,RAIN)
`WSWind`	string	Wind status of Weather control (UNKNOW,CALM,WINDY,VERY_WINDY)
`WSLight`	string	DayLight status of Weather control (UNKNOW,DARK,LIGHT,VERY_LIGHT)

Example:

{"Event":"WeatherAndSafetyMonitorData","Timestamp":1653781759.49165,"Host":"ORIONE","Inst":1,"WSConnected":true,"SMConnected":true,"SMStatus":"SAFE","WSCloud":"CLEAR","WSRain":"DRY","WSWind":"CALM","WSLight":"DARK"}

k) ShotRunning

When a shot start and each 1s after starting and at end of shot Voyager Server send this event to a client of type dashboard.

Attribute	Type	Description
`File`	String	The name (only name) of file running
`Expo`	Number	Exposure lenght of shot in seconds
`Elapsed`	Number	Time elapsed in seconds from start
`ElapsedPerc`	Integer	Percentage of elapsed
`Status`	Integer	Shot Status , see table below

STATUS	Description	Note
0	IDLE	No Exposure
1	EXPOSE	Exposing
2	DOWNLOAD	Download running from camera to PC
3	WAIT_JPG	Process to create a JPG file for Dashboard is running, will finish with a NewJPGReady message
4	ERRORE	Camera Error, shot is aborted

Example:

{"Event":"ShotRunning","Timestamp":1564498706.03752,"Host":"hal9000","Inst":1,"File":"TestShot_20190730_145825.fit","Expo":0.01,"Elapsed":0.01,"ElapsedPerc":100,"Status":1}

l) LogEvent

Report the monitor line of log that showed in Monitor Window in Voyager with possibilities to select the verbose type.

Attribute	Type	Description
TimeInfo	Date	Time of log event write
Type	Integer	Type of event logged
Text	String	Text of event logged

TYPE	Description	Note
1	DEBUG	Low level info
2	INFO	Normal Info
3	WARNING	Warning info
4	CRITICAL	Critical info like an error
5	TITLE	Action running title
6	SUBTITLE	SubAction running title
7	EVENT	Event
8	REQUEST	Command
9	EMERGENCY	Emergency Management

Example:

{"Event":"LogEvent","Timestamp":1564498706.03752,"Host":"hal9000","Inst":1,
,"TimeInfo":1564498706.03752, "Type":1,"Text":"Log line about null nothing" }

m) AutoFocusResult

Contains data about autofocus result just finished in remote server. Sended only to Dashboard client.

Attribute	Type	Description
IsEmpty	Boolean	If false mean no useful data in this event

Done	Boolean	true Focus done (false mean error o not started correctly)
Position	Number	Indicate focuser position in step of autofocus
HFD	Number	Indicate HFD final reched by autofocus
StarPosition	Object	Indicate X and Y in pixel of star centroids used for focus if single star focus
DoneTime	Epoch	Date time of focus
Duration	String	String in mm:ss of duration of autofocus action
FocusTemp	Number	Temperature readed from focuser during the autofocus (°C , °F or ADU depends on your focuser)
PercDev	Number	Deviation from the last series of autofocus if data are available
LastError	String	If focus is not done the text about the reason/error if available
FilterIndex	Number	Index base 0 of filter used for focus, -1 if data on filter is not available
FilterColor	String	Color RGB in HTML format to draw the filter in Graphic GUI (Ex. “#FF0000” = Red, “#FFFFFF” = white. Always “#FFFFFF” if FilterIndex is -1

Example:

{"Event":"AutoFocusResult","Timestamp":1580817847.51588,"Host":"hal9000","Inst":1,"IsEmpty":"false","Done":true,"Position":53149,"HFD":5.00713205337524,"StarPosition":{"X":421,"Y":796},"DoneTime":1580817847.49987,"Duration":"00:00:07","FocusTemp":0.959999978542328,"PercDev":0,"LastError":"","FilterColor":"#FFFFFF",,"FilterIndex":0}

n) ProfileChanged

Contains data about new profile just loaded in Voyager remote server.

Attribute	Type	Description
NewProfile	String	Filename with extension of profile just loaded

Example:

{"Event":"ProfileChanged","Timestamp":1580894669.25674,"Host":"hal9000","Inst":1,"NewProfile":"SimulatoreCorso.v2y"}

o) VikingManaged

Raised when a dashboard mode client connect to Voyager Application Server or if the user change the Viking Manage flag in Voyager Setup and the client is in dashboard mode.

Attribute	Type	Description
IsManaged	boolean	True if is managed , false if not managed
ClientNum	integer	Number of client like configured in Voyager (1 is the first)

Example:

{"Event":"VikingManaged","Timestamp":1604426976.17208,"Host":"ORIONE","Inst":1,"IsManaged":true,"ClientNum":1 }

p) VikingIOConfiguration

Received when Viking connected to Voyager. Raised when a dashboard mode client send the command to activate the rx of All Status Data event coming from Viking. The data in the event is related to the full actual configuration of Viking application connected to the specified client.

Attribute	Type	Description
Out	integer	Number of relays output configured in Viking. 0 is equal to not output
OutConf	Vector of Configuration Objects	List of Outputs configuration: • Index -> Integer -> index of output • Description -> String -> name of the output • Hide -> Boolean -> tell if the output must be showed or not also if is configured
DigIn	integer	Number of Digital Input configured in Viking. 0 is equal to not input
DigInConf	Vector of Configuration Objects	List of digital input configuration: • Index -> Integer -> index of digital input • Description -> String -> name of the digital input • Hide -> Boolean -> tell if the digital input must be showed or not also if is configured
AnaIn	integer	Number of Analog Input configured in Viking. 0 is equal to not input
AnaInMaxValue	integer	Max value for an Analog Input
AnaInConf	Vector of Configuration Objects	List of analog input configuration: • Index -> Integer -> index of analog input • Description -> String -> name of the analog input • Hide -> Boolean -> tell if the analog input must be showed or not also if is configured • PHY -> Boolean -> indicates whether the value represents a physical quantity • PHYLabel -> string -> Label to append to rescaled value text • PHYFactorScale -> double -> value to divide to obtain physycal quantity • PHYNullValue -> double -> value represents a null value
PWM	integer	Number of PWM output configured in Viking. 0 is equal to not output
PWMConf	Vector of Configuration Objects	List of PWM output configuration: • Index -> Integer -> index of PWM output • Description -> String -> name of the PWM output • Hide -> Boolean -> tell if the PWM output must be showed or not also if is configured
DAC	integer	Number of DAC output configured in Viking. 0 is equal to not output
DACMaxValue	integer	Max value for a DAC Output
DACConf	Vector of Configuration Objects	List of DAC output configuration: • Index -> Integer -> index of DAC output • Description -> String -> name of the DAC output • Hide -> Boolean -> tell if the DAC output must be showed or not also if is configured

Automa	integer	Number of Automa output configured in Viking. 0 is equal to not output
AutomaConf	Vector of Configuration Objects	List of Automa output configuration: • Index -> Integer -> index of Automa output • Description -> String -> name of the Automa output • Hide -> Boolean -> tell if the Automa output must be showed or not also if is configured
ClientNum	Integer	The reference of which Viking Client in Voyager have reported the data. This helo to know which is the Viking Server configured with this data

Example:

{"Event":"VikingIOConfiguration","Timestamp":1604931915.66639,"Host":"ORIONE","Inst":1,"Out":4,"OutConf":[{"Index":1,"Description":"Power 1","Hide":false},{"Index":2,"Description":"Power 2","Hide":false},{"Index":3,"Description":"Power 3","Hide":false},{"Index":4,"Description":"Power 4","Hide":false}],"DigIn":2,"DigInConf":[{"Index":1,"Description":"Input 1","Hide":false},{"Index":2,"Description":"Input 2","Hide":false}],"AnaIn":1,"AnaInMaxValue":1024,"AnaInConf":[{"Index":1,"Description":"Input 1","Hide":false}],"PWM":1,"PWMConf":[{"Index":1,"Description":"PWM 1","Hide":false}],"DAC":1,"DACMaxValue":1024,"DACConf":[{"Index":1,"Description":"DAC 1","Hide":false}],"Automa":1,"AutomaConf":[{"Index":1,"Description":"Automation 1","Hide":false}],"ClientNum":1}

q) AllStatus

Raised each 2s if Viking is connected to Voyager and sended to a dashboard mode client connect to Voyager Application Server if the client have activated the viking send data mode

Attribute	Type	Description
Link	String	Stato connessione della Scheda di I/O
Out	integer	Number of status (Relays Output)
OutData	Integer array	Status 0 = OFF, 1 = ON , -1 = UNKNOW
DigIn	integer	Number of status (Digital Input)
DigInData	Integer array	Status 0 = OFF, 1 = ON , -1 = UNKNOW
AnaIn	integer	Number of status (Analog Input)
AnaInData	Integer array	Value (-1 = UNKNOW)
PWM	integer	Number of status (PWM Output)
PWMData	Integer array	Value
DAC	integer	Number of status (DAC Output)
DACData	Integer array	Value (-1 = UNKNOW)
Automa	integer	Number of status (Automa Output)
Automa Data	Integer array	-1 = UNKNOW , 0 = CLOSE , 1 = OPEN, 2 = STOP
ClientNum	Integer	Client where data coming, correspond to the Viking server

Example:

{"Event":"AllStatus","Timestamp":1604932907.50054,"Host":"ORIONE","Inst":1,"Link":"ON","Out":4,"OutData":[1,0,0,1],"DigIn":2,"DigInData":[1,1],"AnaIn":1,"AnaInData":[0],"PWM":1,"PWMData":[0],"DAC":1,"DACData":[0],"Automa":1,"AutomaData":[-1],"ClientNum":1}

r) VikingDisconnected

Raised when Viking Application connected to Voyager is closed or socket disconnected. To all dashboard mode client Voyager will send this event

Attribute	Type	Description
`ClientNum`	integer	Number of client like configured in Voyager (1 is the first)

Example:

{"Event":"VikingDisconnected","Timestamp":1605097803.10557,"Host":"ORIONE","Inst":1,"ClientNum":1}

6. 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. Parameters 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) RemoteActionAbortAll

Method	`RemoteActionAbortAll`
Description	Ask to the Server to abort all the actions running. Do not wait for abort and do not send result of Abort.
Params	UID | String | Unique identifier of the Action. Use a Guide Window identifier or a unique key string generated
Result	Integer(0)
License Required	*Base, Advanced, Full, Custom*
Remote Action Result Parameters	

➔{"method": "RemoteActionAbortAll", "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}

e) RemoteCameraShot

Method	`RemoteCameraShot`
Description	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.
Params	

UID	String	Unique identifier of the Action to abort
Expo	Number	Time of exposure expressed in seconds
Bin	Integer	Binning value for x and y
IsROI	Boolean	true if you want to use some kind of ROI, false for full framing
ROITYPE	Integer	See table below
ROIX	Integer	ROI x origin in pixel
ROIY	Integer	ROI y origin in pixel
ROIDX	Integer	ROI width x in pixel
ROIDY	Integer	ROI width y in pixel
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
ExpoType	Integer	See table of types in `NewFITReady` event
SpeedIndex	Integer	Index of filter to user for exposure like received in `RemoteGetSpeedConfiguration` or 0 for default
ReadoutIndex	Integer	Index of filter to user for exposure like received in `RemoteGetReadoutConfiguration` or 0 for default
IsSaveFile	Boolean	true always
FitFileName	String	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.
Gain	Integer	For CMOS camera, setting the Gain, a SPECIAL VALUES can be used, see table below.

	Offset	Integer	For CMOS camera, setting the Offset, a SPECIAL VALUES can be used, see table below
	Parallelized	Boolean	True if you want to run the remote action in parallel to an actual running local action, default is false. High recommended to use false if is not necessary
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		
Remote Action Result Parameters	DownloadAndSaveTime	Number	Time necessary for download data from camera

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

OFFET AND GAIN SPECIAL VALUES	Description
-2147483648	`GAIN_OFFSET_NULL` - NULL value , Voyager doesn't modify actual Gain used and not track it in log . This value is in C# or VBNet the Integer.MinValue costant
-900000	`GAIN_OFFSET_PRESET` – Preset value, Voyager use the preset Gain value stored in actual Setting Profile
-800000	`GAIN_OFFSET_ACTUAL` – Actual value, Voyager use the actual value in Camera

➔{"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","Gain":78,"Offset":22}, "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}}

f) 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":{}}

g) 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":{}}

h) 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":{}}

i) 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}}

j) 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":{}}

k) RemoteFocus

<table>
<tr><td>Method</td><td>RemoteFocus</td></tr>
<tr><td>Description</td><td>[DEPRECATED] Execute AutoFocus Action in Remote Voyager Server. Reserved to Array Operations, no star goto. Use RemoteFocusEx</td></tr>
<tr><td>Params</td><td>
<table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr>
<tr><td>IsRoboFireLocalField</td><td>Boolean</td><td>true if you want to use the RoboFire LocalField Autofocus on all CCD Frame, false to use RoboFire with RoboStar selection on single star</td></tr>
<tr><td>IsAsyncMode</td><td>Boolean</td><td>Always use true</td></tr>
<tr><td>filtroFuocoIndex</td><td>Integer</td><td>Index of filter to use for focus like received in RemoteGetFiltersConfiguration or 0 for DSLR or COLOR CCD or no filter camera setup</td></tr>
<tr><td>IsWDMaxHFDVariation</td><td>Boolean</td><td>true if you want repeat focus if result HFD is greater than a certain variation value in percentage in the last autofocus HFD mobile mean</td></tr>
<tr><td>WDMaxHFDLimitVariation</td><td>Number</td><td>Max value percentage of HFD variation considered good</td></tr>
<tr><td>IsWDMaxHFD</td><td>Boolean</td><td>true if you want repeat focus if result HFD is great than a certain value in pixel</td></tr>
<tr><td>WDMaxHFDLimit</td><td>Number</td><td>Max value in pixel of final HFD considered good</td></tr>
<tr><td>IsRetryFocusOnWD</td><td>Boolean</td><td>true to retry autofocus if ones of the WD is happen or false to return to previous focus position</td></tr>
<tr><td>PreviousPosition</td><td>Integer</td><td>Value in step of previous focus position</td></tr>
<tr><td>IsFMAcquireStarFocus</td><td>Boolean</td><td>true if you want to use FocusMax Acquire Star routine (use false to use Voyager RoboStar)</td></tr>
<tr><td>StarName</td><td>String</td><td>Name of focus star If you want to use a defined star for Focus on Star mode</td></tr>
</table>
</td></tr>
<tr><td>Result</td><td>Integer(0)</td></tr>
<tr><td>License Required</td><td>Base, Advanced, Full, Custom</td></tr>
</table>

➔{"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":{}}

l) RemoteFocusEx

Method	`RemoteFocusEx`
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
FocusMode	Integer	See table below
filtroFuocoIndex	Integer	Index of filter to use for focus like received in `RemoteGetFiltersConfiguration` or 0 for DSLR or COLOR CCD or no filter camera setup
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 (-1 if you don't know)
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 (-1 if you don't know)
StarRAJ2000Str	String	RA coordinate in J2000 string format HH MM SS.sss of the star to use for focus
StarDECJ2000Str	String	DEC coordinate in J2000 string format HH MM SS.sss of the star to use for focus
IsGoBack	Boolean	ONLY If you used a focus method that do a goto to star : true if you want to come back to original position at finished focus or false if you want to remain on focus star
IsOnlyPointingStar	Boolean	ONLY If you used a focus method that do a goto to star : True if you want just to move to the focus star, false in other cases

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

FocusMode	Description

0	**(Focus Star)** Focus on Star choose by StarName parameter. A precise goto will be done to the star for focus
1	**(AcquireStar FM)** Focus with FocusMax AcquireStar facilities, FocusMax must be installed and configured correctly
2	**(On Place)** No goto , just a focus will be tried on the place where is the telescope, lucky mode or you already pointed to a right focus star
3	**(Voyager RoboStar)** Focus on a star choosed by RoboStar.. A precise goto will be done to the focus star according Filter parameters
4	**(Voyager LocalField)** focus on full frame using Voyager LocalField AI
5	**(Only Pointing with RoboStar)** A precise goto will be done to the focus star. Nothing else.

➔{"method": "RemoteFocusEx", "params": {"UID":"dd486bd0-b141-43e8-a401-4871cea992f4","FocusMode":0,"filtroFuocoIndex":1,"IsWDMaxHFD":false,"WDMaxHFDLimit":9.4,"IsRetryFocusOnWD":true,"PreviousPosition":-1,"StarRAJ2000Str":"11 22 32.123","StarDECJ2000Str":"22 11 04.123","IsGoBack":true, "IsOnlyPointingStar":false }, "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":{}}

m) RemoteFocusInject

Method	`RemoteFocusInject`
Description	Inject a remote Autofocus in running Sequence (if there is one) 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 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
Result	Integer(0)
License Required	*Base, Advanced, Full, Custom*

➔{"method": "RemoteFocusInject", "params": {"UID":"dd486bd0-b141-43e8-a401-4871cea992f4","filtroFuocoIndex":1}, "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":"Cannot inject focus no Sequence running")","ParamRet":{}}

n) 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":{}}

o) 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":{}}

p) RemoteGetStatus

Method	`RemoteGetStatus`
Description	Return Operative Status of Voyager Application
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	VoyagerStatus (String) — Voyager Operative Status, see the dedicated table

VoyagerStatus	Description
STOPPED	Application is Stopped, Profile is disconnected
IDLE	Application with Profile connected and in IDLE (no action running)
RUN	An Action is running
ERRORE	Last Action run thrown an ERROR
UNDEFINED	Unknow status
WARNING	Last Action run thrown a WARNING

➔ {"method": "RemoteGetStatus", "params": {"UID":"47a439a9-6453-477c-b5c4-529a93605867"}, "id": 369}

➔{"jsonrpc": "2.0", "result": 0, "id": 369}

➔
{"Event":"RemoteActionResult","Timestamp":1666462325.20341,"Host":"ORIONE","Inst":1,"UID":"47a439a9-6453-477c-b5c4-529a93605867","ActionResultInt":4,"Motivo":"","ParamRet":{"VoyagerStatus":"RUN"}}

q) 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}}

r) 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}}

s) 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}}

t) 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}}

u) 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":{}}

v) RemoteRotatorSync

Method	`RemoteRotatorSync`		
Description	Sync the rotator to the PA requested in Remote Voyager Server, or reset it to the mechanical position		
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
	IsReset	Boolean	true if you want to reset the sync to mechanical position
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

→{"method": "RemoteRotatorSync", "params": {"UID":"a53c6e8a-be1d-4c67-8ed7-df41c15d8923","PA":0,"IsReset":false}, "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":{}}

w) 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":{}}

x) 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":{}}

y) 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":{}}

z) 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}}

aa) 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}}

bb) 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}}

cc)RemoteSetDashboardMode

Method	RemoteSetDashboardMode		
Description	When the client connect to Application Server can specify if is a Dashboard client or normal client with this command. If a client is a Dashboard , the Application Server will send a NewJPGReady event when a new image will be ready on the disk. This event will contain the base64 data of the fit image stretched and compressed in JPG quality		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	IsOn	Boolean	true if the client will be a dashboard
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

➔ {"method": "RemoteSetDashboardMode", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","IsOn":true}, "id": 2}

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

🡐{"Event":"RemoteActionResult","Timestamp":1556990521.31099,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{}}

dd) RemoteGetListAvalaibleSequence

Method	RemoteGetListAvalaibleSequence		
Description	Retrieve list of sequence file (with extension) in Remote Default directory of Voyager		
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	List	Array String	Array of name of all Sequence files founded in Sequence default directory of remote Voyager in alphabetical order

➔ {"method": "RemoteGetListAvalaibleSequence", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8"}, "id": 2}

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

←{"Event":"RemoteActionResult","Timestamp":1562942486.31045,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{"list":["aa.s2q","CalibrationSequence.s2q","ddd.s2q","ee.s2q","eeee.s2q","FoxFurGenerica.s2q","LBN438_Col.s2q","LBN438_Lum.s2q","LDN183_L.s2q","LDN183_RGB.s2q","M100L.s2q","M33-mosaico4-L.s2q","M63.s2q","M97.s2q","NGC1788.s2q","NGC2170_L.s2q","NGC2170_RGB.s2q","NGC2683_Rila600.s2q","peppa.s2q","pippo.s2q","PLN164_LRGB.s2q","Profilo Test2.s2q","Rila_IC417_12minHa.s2q","Rila_IC417_5minHa.s2q","seqr.s2q","SequenzaTestRelease.s2q","testDefaultSeq.s2q","TestFuocoalMeridiano.s2q","TestGuided.s2q","TestM13.s2q","TestMeridianCheck.s2q","TestNoCalibra.s2q","TestNoPlateSolving.s2q","TestOnlyExpoProb.s2q","TestRoboGuide.s2q","TestSeqWithScript.s2q","TestUnguided.s2q","TestUnguidedNoPlateSolve.s2q","zumba.s2q"]}}

ee) RemoteGetListAvalaibleSequenceEx

<table>
<tr><td>Method</td><td colspan="4">RemoteGetListAvalaibleSequenceEx</td></tr>
<tr><td>Description</td><td colspan="4">Return list of Sequence available for a profile in default Voyager Sequence configuration folder</td></tr>
<tr><td>Params</td><td colspan="4"><table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr>
<tr><td>ProfileName</td><td>String</td><td>Profile name to use for the search. If empty return all the sequence in the sequence folder</td></tr>
</table></td></tr>
<tr><td>Result</td><td colspan="4">Integer(0)</td></tr>
<tr><td>License Required</td><td colspan="4">Base, Advanced, Full, Custom</td></tr>
<tr><td>Remote Action Result Parameters in ParamRet Object</td><td>List</td><td>Array</td><td colspan="2">Array of Sequence Objects<table>
<tr><td>name</td><td>String</td><td>Sequence name</td></tr>
<tr><td>filename</td><td>string</td><td>Sequence file with path</td></tr>
<tr><td>profilename</td><td>string</td><td>Profile name associated to the sequence</td></tr>
</table></td></tr>
</table>

(*) hash reported in the example are only for didattical scope and the final MAC are not correct

→ {“method”: “RemoteGetListAvalaibleSequenceEx”, “params”: {“ProfileName”:”TestFlatNoMount.v2y”,”UID”:”98129170-e267-4f8b-9°21-4e773b2889de”}, “id”: 22}

←{“jsonrpc”: “2.0”, “result”: 0, “id”: 22}

←

{“Event”:”RemoteActionResult”,”Timestamp”:1652626905.66952,”Host”:”ORIONE”,”Inst”:1,”UID”:”98129170-e267-4f8b-9°21-4e773b2889de”,”ActionResultInt”:4,”Motivo”:””,”ParamRet”:{“list”:[{ “name”:”SequenzaBase_TestFlatNoMount.s2q”,

“filename”:”C:\\Users\\pegas\\OneDrive\\Documenti\\Voyager\\ConfigSequence\\SequenzaBase_TestFlatNoMount.s2q”, “profilename”:”TestFlatNoMount.v2y” },{ “name”:”TestRotatoreMeridiano.s2q”, “filename”:”C:\\Users\\pegas\\OneDrive\\Documenti\\Voyager\\ConfigSequence\\TestRotatoreMeridiano.s2q”, “profilename”:”TestFlatNoMount.v2y” },{ “name”:”TestUnguidedNoPlateSolve.s2q”, “filename”:”C:\\Users\\pegas\\OneDrive\\Documenti\\Voyager\\ConfigSequence\\TestUnguidedNoPlateSolve.s2q”, “profilename”:”TestFlatNoMount.v2y” }]}}

ff) RemoteGetListAvalaibleDragScript

<table>
<tr><td>Method</td><td colspan="3">RemoteGetListAvalaibleDragScript</td></tr>
<tr><td>Description</td><td colspan="3">Retrieve list of DragScript file (with extension) in Remote Default directory of Voyager</td></tr>
<tr><td>Params</td><td colspan="3"><table><tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr></table></td></tr>
<tr><td>Result</td><td colspan="3">Integer(0)</td></tr>
<tr><td>License Required</td><td colspan="3">Base, Advanced, Full, Custom</td></tr>
<tr><td>Remote Action Result Parameters</td><td colspan="3"><table><tr><td>List</td><td>Array String</td><td>Array of name of all DragScript files founded in Script default directory of remote Voyager in alphabetical order</td></tr></table></td></tr>
</table>

➔ {"method": "RemoteGetListAvalaibleDragScript", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8"}, "id": 2}

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

←{"Event":"RemoteActionResult","Timestamp":1567252762.08582,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{"list":["aa.vos","AttesaAltitudineBlocco.vos","Bias-Dark - 15T.vos","CalibrationFITandFLAT.vos","CalibrationFITandFLAT_short.vos","dd.vos","debugExpoBefore.vos","decimali.vos","DEmo1.vos","DemoFlatAndOtherCalibrationFile.vos","DemoMultiSequenceNight.vos","dscript 1.vos","dscript1.vos","dscript2.vos","EmergencyExit.vos","EmergencyExitDefault.vos","EnableDisableEventi.vos","esposizione-multipla- 1sec.vos","FDOpen.vos","FDOpen_OLD.vos","FitCalibrazione.vos","FlatConDusk.vos","ForWayne.vos","FS2OutOfPark.vos","FullNight-2019-07-14.vos","JoachimCloudWatcher.vos]}}

gg) RemoteSetLogEvent

Method	RemoteSetLogEvent
Description	Ask to Server to send or not send log events from Voyager. Log Events is what reported in Monitor Window of Voyager. You can activate and deactivate and decide level of details in logging
Params	

UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
IsOn	Boolean	true if the log events will be sended
Level	Integer	Level of details in log. 0 = All ; 1=Only emergency,critical,warning,event,title,subtitle (Info and debug will be removed)

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

➔ {"method": "RemoteSetLogEvent", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","IsOn":true, "Level":0}, "id": 2}

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

←{"Event":"RemoteActionResult","Timestamp":1556990521.31099,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{}}

hh) RemoteSearchTarget

Method	RemoteSearchTarget
Description	Search a Target from a Planetarium connected to Voyager or to Simbad online
Params	

UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
Name	String	Target name to search
SearchType	Integer	Search type. 0 = Planetarium ; 1=Simbad For Simbad search this will be done using remote Voyager no directly from client

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

Result	Integer	0=NOT FOUND 1=FOUND 2=ERROR
LastError	String	If an error thrown the text of error or empty
Name	String	Name normalized of Target found
RAJ2000	String	RA coord in string format in J2000 epoch
DECJ2000	String	DEC coord in string format in J2000 epoch

	Info	Array	Array of object made by key (string) and value (string) Key = name of info Value = value of info

➔ {"method": "RemoteSearchTarget", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","Name":"M31", "SearchType":0}, "id": 2}

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

🡐{"Event":"RemoteActionResult","Timestamp":1564605292.52132,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{"Result":1,"LastError":"","Name":"M31","RAJ2000":"00 42 41,582","DECJ2000":"41 15 59,97","Info":[{"Key":"Charts ID","Value":"Cartina_1"},{"Key":"AR(JNow)","Value":"00h43m46.41s"},{"Key":"DEC(JNow)","Value":"+41\u00B022'11.0\""},{"Key":"Type","Value":"Gx"},{"Key":"Name","Value":"M31"},{"Key":"m","Value":"3.40"},{"Key":"Name","Value":"NGC 224"},{"Key":"sbr","Value":"13.50"},{"Key":"Dim","Value":"189.0 x 61.0'"},{"Key":"pa","Value":"35"},{"Key":"class","Value":"Sb"},{"Key":"desc","Value":"!!!eeB;eL;vmE;Local Group;Andromeda Galaxy;nearest spiral"},{"Key":"Const","Value":"AND"}]}}

ii) RemoteGetEnvironmentData

Method	`RemoteGetEnvironmentData`
Description	Retrieve Actual Profile information of Voyager in terms of profile name and controls type
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	Profile — String — Type of Control Camera — String — “ FilterWheel — String — “ Mount — String — “ Guide — String — “ Planetarium — String — “ PlateSolve — String — “ BlindSolve — String — “ Focuser — String — “ AutoFocus — String — “ Rotator — String — “ FlatDevice1 — String — “ FlatDevice2 — String — “

	Dome	String	“
	ObservingConditions	String	“
	SQM	String	“
	SafetyMonitor	String	“

➔ {"method": "RemoteGetEnvironmentData", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8"}, "id": 2}

➔{"jsonrpc": "2.0", "result": 0, "id": 2}

➔ {"Event":"RemoteActionResult","Timestamp":1564750461.29792,"Host":"hal9000","Inst":1,"UID": "eaea5429-f5a9-4012-bc9b-f109e605f5d8", "ActionResultInt":4,"Motivo":"","ParamRet":{"Profile":"TestASINative","Camera":"ASCOM Camera [ASCOM.Simulator.Camera]","FilterWheel":"ASCOM Filter Wheel [FilterWheelSim.FilterWheel]","Mount":"ASCOM Mount [ScopeSim.Telescope]","Guide":"PHD2 Guide","Planetarium":"","PlateSolve":"PlateSolve2","BlindSolve":"","Focuser":"","AutoFocus":"","Rotator":"","FlatDevice1":"","FlatDevice2":"","Dome":"","ObservingConditions":"","SQM":"","SafetyMonitor":""}}

jj) Abort

Method	`Abort`
Description	Abort actual action running in Voyager or HALT ALL activities (also action) running in Voyager
Params	IsHalt — Boolean — true for HALT ALL, false for just abort actual running action
Result	Integer(0) – **NO RemoteActionResult for this Command**
License Required	*Base, Advanced, Full, Custom*

➔ {"method": "Abort", "params": {"IsHalt":false}, "id": 2}

➔{"jsonrpc": "2.0", "result": 0, "id": 2}

kk) RemotePulseGuide

Method	`RemotePulseGuide`
Description	Move the mount using the Pulse Guide method, mount must be out of park and tracking and able to use Pulse Guide at driver level.
Params	

	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	Direction	Integer	See the direction table below for values
	Duration	Integer	Time in milliseconds of the pulse
	Parallelized	Boolean	True if you want to run the remote action in parallel to an actual running local action, default is false. High recommended to use false if is not necessary
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

Direction	Description
0	guideNorth North (+ declination/altitude).
1	guideSouth South (- declination/altitude).
2	guideEast East (+ right ascension/azimuth).
3	guideWest West (- right ascension/azimuth)

➔ {"method": "RemotePulseGuide", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","Direction":0,"Duration":1250}, "id": 2}

🡐{"jsonrpc": "2.0", "result": 0, "id": 2}

🡐 {"Event":"RemoteActionResult","Timestamp":1567083112.28221,"Host":"hal9000","Inst":1, "UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"" ,"ParamRet":{}}

ll) RemoteGotoAltAz

Method	`RemoteGotoAltAz`		
Description	Pointing a Target with Altitude Azimuth coords		
Params			
	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	ALT	Numeric	Altitude in Degree
	AZ	Numeric	Azimuth in Degree
	SettleTime	Integer	Time in seconds to wait after the goto is finished to allow mount to settle micro movements if necessary
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

➔ {"method": "RemoteGotoAltAz", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ALT":12.0,"AZ":11.0,"SettleTime":5}, "id": 2}

🡐{"jsonrpc": "2.0", "result": 0, "id": 2}

🡐 {"Event":"RemoteActionResult","Timestamp":1567083112.28221,"Host":"hal9000","Inst":1, "UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"" ,"ParamRet":{}}

mm) RemotePrecisePointTarget

Method	RemotePrecisePointTarget
Description	Pointing a Target with coords text or double in a precise way . Coords must be in J2000
Params	

UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
IsText	Boolean	true if coord are in text format or false if coord are in double (hour and degree) format
RA	Numeric	0 or value in hour and decimal
DEC	Numeric	0 or value in degree and decimal
RAText	String	String HH MM SS.SSS or empty
DECText	String	String DD MM SS.SSS or empty
Parallelized	Boolean	True if you want to run the remote action in parallel to an actual running local action, default is false. High recommended to use false if is not necessary

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

ActionResult	Integer	Return of action in details see table below

ActionResult	Description
0	FAILED
1	OK IN RANGE
2	OK OUT OF RANGE
3	OK PLATE SOLVING DISABLED

➔ {"method": "RemotePrecisePointTarget", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","IsText":false,"RA":12.0,"DEC":11.0,"RAText":"","DECText":""}, "id": 2}

🡐{"jsonrpc": "2.0", "result": 0, "id": 2}

← {"Event":"RemoteActionResult","Timestamp":1567083112.28221,"Host":"hal9000","Inst":1, "UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":5,"Motivo":"Blind Solving Control is Empty","ParamRet":{"ActionResult":0}}

nn) RemotePrecisePointTargetAndPA

<table>
<tr><td>Method</td><td>RemotePrecisePointTargetAndPA</td></tr>
<tr><td>Description</td><td>Pointing a Target with coords text or double in a precise way and using the Rotator to match the specified PA. Coords must be in J2000</td></tr>
<tr><td>Params</td><td>
<table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr>
<tr><td>IsText</td><td>Boolean</td><td>true if coord are in text format or false if coord are in double (hour and degree) format</td></tr>
<tr><td>RA</td><td>Numeric</td><td>0 or value in hour and decimal</td></tr>
<tr><td>DEC</td><td>Numeric</td><td>0 or value in degree and decimal</td></tr>
<tr><td>RAText</td><td>String</td><td>String HH MM SS.SSS or empty</td></tr>
<tr><td>DECText</td><td>String</td><td>String DD MM SS.SSS or empty</td></tr>
<tr><td>PA</td><td>Numeri</td><td>Value of Target Position Angle in Degree</td></tr>
<tr><td>PATollerance</td><td>Numeri</td><td>Tollerance +/- in Degree between the asked PA and PA interval considered accepted without move rotator</td></tr>
<tr><td>IsSkyPA</td><td>Boolean</td><td>true if the action using the Sky PA mode and try to align rotator using the PA value solved on the sky from Plate/Blind Solving, false to use the rotator PA without check on Sky PA</td></tr>
<tr><td>MantainImageOrientation</td><td>Boolean</td><td>true if you want to maintain image orientation when the mount is after the meridian . Depending if you using sky PA o rotator PA a rotator flip will be done. For example if you want to use the same guide star after meridian , put this properties to true.</td></tr>
</table>
</td></tr>
<tr><td>Result</td><td>Integer(0)</td></tr>
<tr><td>License Required</td><td>Base, Advanced, Full, Custom</td></tr>
<tr><td>Remote Action Result Parameters</td><td>
<table>
<tr><td>ActionResult</td><td>Integer</td><td>Return of action in details see table below</td></tr>
</table>
</td></tr>
</table>

ActionResult	Description
0	FAILED

1	OK IN RANGE
2	OK OUT OF RANGE
3	OK PLATE SOLVING DISABLED

➔ {"method": "RemotePrecisePointTargetAndPA", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","IsText":false,"RA":12.0,"DEC":11.0,"RAText":"","DECText":"","PA":123.12,"PATollerance":3.0,"IsRotatorSync":false,"IsPAAllow180":false}, "id": 2}

➔{"jsonrpc": "2.0", "result": 0, "id": 2}

➔ {"Event":"RemoteActionResult","Timestamp":1567083112.28221,"Host":"hal9000","Inst":1, "UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":5,"Motivo":"Blind Solving Control is Empty","ParamRet":{"ActionResult":0}}

oo) RemoteSequence

Method	`RemoteSequence`
Description	Execute a Sequence in the remote Voyager
Params	

UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
SequenceFile	String	Name of the file with extension to run. File must be placed in the default sequence directory in Voyager. Personalized path are not allowed
StartFlag	Integer	Startup flag, see table below for list of values . Flag can be sum togheter to give multi choices (example 15 is equal to all possibilities)

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

StartFlag	Description
0	NORMAL
1	REMOVE Initial Precise Pointing
2	REMOVE Initial Focus
4	REMOVE Guide Calibration
8	REMOVE Precise Pointing before first shot

➔ {"method": "RemoteSequence", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","SequenceFile":"TestUnguidedNoPlateSolve.s2q","StartFlag":0}, "id": 2}

🡐{"jsonrpc": "2.0", "result": 0, "id": 2}

🡐 {"Event":"RemoteActionResult","Timestamp":1567083112.28221,"Host":"hal9000","Inst":1, "UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{}}

pp) RemoteDragScript

<table>
<tr><td>Method</td><td colspan="3">RemoteDragScript</td></tr>
<tr><td>Description</td><td colspan="3">Execute a DragScript in the remote Voyager</td></tr>
<tr><td>Params</td><td colspan="3"><table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr>
<tr><td>DragScriptFile</td><td>String</td><td>Name of the file with extension to run. File must be placed in the default script directory in Voyager. Personalized path are not allowed</td></tr>
<tr><td>StartNodeUID</td><td>String</td><td>If not empty indicate a node to execute like first.</td></tr>
</table></td></tr>
<tr><td>Result</td><td colspan="3">Integer(0) – This Comand doesn’t return a RemoteActionResult check the Result if an error is occurred in starting script</td></tr>
<tr><td>License Required</td><td colspan="3">Base, Advanced, Full, Custom</td></tr>
</table>

➔ {"method": "RemoteDragScript", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","DragScriptFile":"Pippo.s2q","StartNodeUID":""}, "id": 2}

🡐{"jsonrpc": "2.0", "result": 0, "id": 2}

qq) RemoteDragScriptSelfContained

Method	RemoteDragScriptSelfContained
Description	Execute a DragScript in the remote Voyager and wait for finish. Execution of DragScript with this command is synched and the RemoteActionResult will be generated. Self Contained means the DragScript will be executed like all the normal actions and no interaction will be done with the DragScript Session window of Voyager. Use for small operations like set of more action to do with a precise scope, like open a roof, prepare the flat device and mount et similar. Do not run long DragScript because you will not see the status of DragScript running (which line is running etc etc)

	If you use the only name of dragscript, the script will be searched in the default script directory of Voyager otherwise will be loaded directly the file reported. The path to use is the path of the PC where is the script.
Params	<table><tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr><tr><td>DragScriptFile</td><td>String</td><td>Name of the file with extension to run. File must be placed in the default script directory in Voyager. Personalized path are not allowed</td></tr></table>
Result	Integer(0)
License Required	*Base, Advanced, Full, Custom*

➔ {"method": "RemoteDragScript", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","DragScriptFile":"Pippo.s2q","StartNodeUID":""}, "id": 2}

🡐{"jsonrpc": "2.0", "result": 0, "id": 2}

🡐 {"Event":"RemoteActionResult","Timestamp":1567083112.28221,"Host":"hal9000","Inst":1, "UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{}}

rr) RemoteDragScriptSelfContained

Method	`RemoteDragScriptSelfContained`
Description	Execute a DragScript in the remote Voyager and wait for finish. Execution of DragScript with this command is synched and the RemoteActionResult will be generated. Self Contained means the DragScript will be executed like all the normal actions and no interaction will be done with the DragScript Session window of Voyager. Use for small operations like set of more action to do with a precise scope, like open a roof, prepare the flat device and mount et similar. Do not run long DragScript because you will not see the status of DragScript running (which line is running etc etc) If you use the only name of dragscript, the script will be searched in the default script directory of Voyager otherwise will be loaded directly the file reported. The path to use is the path of the PC where is the script.
Params	<table><tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr><tr><td>DragScriptFile</td><td>String</td><td>Name of the file with extension to run. File must be placed in the default script</td></tr></table>

			directory in Voyager. Personalized path are not allowed
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

➔ {"method": "RemoteDragScript", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","DragScriptFile":"Pippo.s2q","StartNodeUID":""}, "id": 2}

←{"jsonrpc": "2.0", "result": 0, "id": 2}

← {"Event":"RemoteActionResult","Timestamp":1567083112.28221,"Host":"hal9000","Inst":1, "UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{}}

ss) RemoteMountFastCommand

Method	`RemoteMountFastCommand`		
Description	Execute a fast command dedicated to Mount: homing, park, unpark, track on, track off, goto near zenith in the remote Voyager		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	CommandType	Integer	Command type, see table below for available commands
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

CommandType	Description
1	Track On
2	Track Off
3	Park
4	Unpark
5	Goto Near Zenith
6	Home

➔ {"method": "RemoteMountFastCommand", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","CommandType":1}, "id": 2}

←{"jsonrpc": "2.0", "result": 0, "id": 2}

← {"Event":"RemoteActionResult","Timestamp":1567083112.28221,"Host":"hal9000","Inst":1, "UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{}}

tt) RemoteGetVoyagerProfiles

Method	`RemoteGetVoyagerProfiles`		
Description	Retrieve the list of Voyager Setup Profile that exists in the Profile directory of Voyager		
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	List	Array Object	Array of object with guid,name,if active or not of all Profile files founded in Profile default directory of remote Voyager in alphabetical order with extension

➔ {"method":"RemoteGetVoyagerProfiles","params":{"UID":"208BBAA7-218D-2B92-B648-B9FFBFCB04F1"},"id":6}

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

← {"Event":"RemoteActionResult","Timestamp":1588508648.91063,"Host":"osservatorio-PC","Inst":1,"UID":"208BBAA7-218D-2B92-B648-B9FFBFCB04F1","ActionResultInt":4,"Motivo":"","ParamRet":{"list":[{"guid":"77b88760-c5fe-4a1a-890f-795a0a420124", "name":"Default.v2y", "isactive":false},{"guid":"9963c012-4ffc-4732-9e9e-4f191da5b329", "name":"Postazione1.v2y", "isactive":false},{"guid":"ece6e864-7f4e-4664-9fa8-27200c804a5c", "name":"Postazione1_provvisoria.v2y", "isactive":false},{"guid":"aa80a367-bd8f-40af-9e43-43652b8459af", "name":"Simulator RC12 Kai4022.v2y", "isactive":true},{"guid":"7324845f-b076-4f03-a560-63dc58ddeb99", "name":"Sim_ RC12 Kai4022.v2y", "isactive":false},{"guid":"c7a52f13-6612-4f7c-acd0-f0319ad3ecad", "name":"test.v2y", "isactive":false}]}}

uu) RemoteSetProfile

Method	`RemoteSetProfile`		
Description	Load Profile in remote in Remote Voyager Server. Work only if Remote Voyager is not connected to any profile		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated

	FileName	String	Profile Name with extension. Profile must reside in the default profile directory of Voyager installation
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

➔{"method": "RemoteSetProfile", "params": {"UID":"a53c6e8a-be1d-4c67-8ed7-df41c15d8923","FileName":"SoloCamera.v2y" }, "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":{}}

Another example when Voyager have a profile already connected:

➔ {"method": "RemoteSetProfile", "params": {"UID":"a53c6e8a-be1d-4c67-8ed7-df41c15d8923","FileName":"SimulatoreCorso.v2y" }, "id": 19423}

➔{"jsonrpc":"2.0","error":{"code":1,"message":"Another Profile is actually connected"},"id":19423}

vv) RemoteGetCCDConfiguration

Method	`RemoteGetCCDConfiguration`		
Description	Return data about CCD (color mode, gain capability and offset capability) 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	IsBayerCamera	boolean	true if camera have a bayer matrix sensor
	HaveGainCapability	boolean	true if camera can set the numeric gain
	HaveOffsetCapability	boolean	true if camera can set the numeric offset

➔{"method": "RemoteGetCCDConfiguration", "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": {" IsBayerCamera ":true,"HaveGainCapability":true," HaveOffsetCapability ":true }}

ww) RemoteSetVikingDataSend

Method	`RemoteSetVikingDataSend`		
Description	Used to tell to remote Voyager connected if send or not the Viking data about list of I/O status. The data will be send periodically each 2 seconds if Viking client is configured in Voyager profile.		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	IsOn	Boolean	true to receive data , false to stop receive data
	ClientNum	Integer	Number of Client where apply the flag, 1 is the first
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

→ {"method": "RemoteSetVikingDataSend", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","IsOn":true,"ClientNum":1 }, "id": 2}

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

←{"Event":"RemoteActionResult","Timestamp":1556990521.31099,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{}}

xx) AuthenticateUserTicket

Method	`AuthenticateUserTicket`		
Description	Authenticate renting user in Voyager, error result will close the connection immediatly		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	Ticket	String	Information Reserved to NDA and agreement. Please ask to Voyager support for a contact.

Result	Integer(0)
License Required	*Base, Advanced, Full, Custom*
Plugin Required	*Renting*

CMD authticket result pameters

ActivationCode	string	Code of the purchased reservation
UserCode	string	Code of the user for unique identify
FirstName	string	First name of renting user
LastName	string	Last name of renting user
RentingStart	datetime	Init of reservation
RentingEnd	datetime	End of reservation
TelescopeStationName	string	Name of the telescope station
TelescopeStationCode	string	Code of the telescope station
PermissionsA	integer	Permission associated to the user. Information Reserved to NDA and agreement. Please ask to Voyager support for a contact.
PermissionsB	integer	Permission associated to the user. Information Reserved to NDA and agreement. Please ask to Voyager support for a contact.
RenterCode	string	Code of the telescope Renter
RenterContactName	string	Who are attendant for the telescope station in case of help
RenterContactMail	string	Mail for contact the attendant
RenterContactSkype	string	Skype contact of the attendant
RenterContactPhone	string	Phone contact of the attendant
Note	string	Note for the renting user

➔ {"method": "AuthenticateUserTicket", "params": {"UID":"37f4962a-73c5-44f5-80e1-d29f029f49a9","Ticket":"xxxxxxxxxxxxxxxxxxxxxxx"}, "id": 84}

Error result:

➔ {"jsonrpc":"2.0","error":{"code":1,"message":"Your reservation is expired"},"id":84}

OK result:

➔

{"jsonrpc":"2.0","authticket":{"UserCode":"U0001","ActivationCode":"A89349002FRT22","FirstName":"Peppino","LastName":"Di Capri","RentingStart":1607340960,"RentingEnd":1638879004,"TelescopeStationName":"Tecnosky 100Q - QHY600","TelescopeStationCode":"P001","Permissions":178293,"RenterCode":"BIGRENT","RenterContactName":"Mario Rossi","RenterContactMail":"mario.rossi@renter.com","RenterContactSkype":"skype@renter.com","RenterContactPhone":"+3901198989893","Note":"This is remote renter user"},"id":84}

yy) AuthenticateUserBase

<table>
<tr><td>Method</td><td>AuthenticateUserBase</td></tr>
<tr><td>Description</td><td>Authenticate remote user in Voyager, error result will close the connection immediatly</td></tr>
<tr><td>Params</td><td>
<table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr>
<tr><td>Base</td><td>String</td><td>Base authentication string constructed as follows:

1. The username and password are combined with a single colon (:). This means that the username itself cannot contain a colon.
2. The resulting string is encoded into an octet sequence. The character set to use for this encoding is by default unspecified, as long as it is compatible with US-ASCII, but the server may suggest use of UTF-8 by sending the charset parameter
3. The resulting string is encoded using Base64</td></tr>
</table>
</td></tr>
<tr><td>Result</td><td>Integer(0)</td></tr>
<tr><td>License Required</td><td>Base, Advanced, Full, Custom</td></tr>
<tr><td>CMD authbase result pameters</td><td>
<table>
<tr><td>Username</td><td>string</td><td>Username of the user for unique identify</td></tr>
<tr><td>FirstName</td><td>string</td><td>First name of remote user</td></tr>
<tr><td>LastName</td><td>string</td><td>Last name of remote user</td></tr>
<tr><td>Mail</td><td>string</td><td>Mail of remote user</td></tr>
<tr><td>PermissionsA</td><td>integer</td><td>Permission associated to the user. Information Reserved to NDA and agreement. Please ask to Voyager support for a contact.</td></tr>
<tr><td>PermissionsB</td><td>integer</td><td>Permission associated to the user. Information Reserved to NDA and agreement. Please ask to Voyager support for a contact.</td></tr>
<tr><td>Note</td><td>string</td><td>Note for the remote user</td></tr>
</table>
</td></tr>
</table>

➔ {"method": "AuthenticateUserBase", "params": {"UID":"37f4962a-73c5-44f5-80e1-d29f029f49a9","Base":"YWRtaW46cGFzc3dvcmQ="}, "id": 84}

Error result:

← {"jsonrpc":"2.0","error":{"code":1,"message":"Authentication Rejected"},"id":84}

OK result:

←
{"jsonrpc":"2.0","authbase":{"Username":"admin","FirstName":"Mario","LastName":"Rossi","Mail":"mario.rossi@mail.com","Permissions":934838,"Note":"Remote User"},"id":84}

zz)RemoteVikingSetOut

Method	RemoteVikingSetOut		
Description	Change OUT digital status using Viking Client		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	ClientNum	Integer	Number of Client where apply the command, 1 is the first
	OutNumber	Integer	Number of digital out
	State	String	Status to set ON or OFF
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

→ {"method": "RemoteVikingSetOut", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ClientNum":1,"OutNumber":1,"State":"ON"}, "id": 2}

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

←{"Event":"RemoteActionResult","Timestamp":1556990521.31099,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{}}

aaa) RemoteVikingSetPWM

Method	RemoteVikingSetPWM		
Description	Change PWM value using Viking Client		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	ClientNum	Integer	Number of Client where apply the command, 1 is the first
	PWMNumber	Integer	Number of PWM
	Value	Integer	Value to set

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

➔ {"method": "RemoteVikingSetPWM", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ClientNum":1,"PWMNumber":1,"Value":35}, "id": 2}

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

←{"Event":"RemoteActionResult","Timestamp":1556990521.31099,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{}}

bbb) RemoteVikingSetDAC

Method	RemoteVikingSetDAC		
Description	Change DAC value using Viking Client		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	ClientNum	Integer	Number of Client where apply the command, 1 is the first
	DACNumber	Integer	Number of DAC
	Value	Integer	Value to set
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

➔ {"method": "RemoteVikingSetDAC", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ClientNum":1,"DACNumber":1,"Value":35}, "id": 2}

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

←{"Event":"RemoteActionResult","Timestamp":1556990521.31099,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{}}

ccc) RemoteVikingSetAutoma

Method	RemoteVikingSetAutoma
Description	Change Automa Out status using Viking Client
Params	

	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	ClientNum	Integer	Number of Client where apply the command, 1 is the first
	OutNumber	Integer	Number of Automa out
	State	String	Value to set OPEN or CLOSE or STOP
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

➔ {"method": "RemoteVikingSetAutoma", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ClientNum":1,"OutNumber":1,"State":"OPEN"}, "id": 2}←{"jsonrpc": "2.0", "result": 0, "id": 19423}

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

←{"Event":"RemoteActionResult","Timestamp":1556990521.31099,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{}}

ddd) RemoteFlatDeviceCMD

Method	`RemoteFlatDeviceCMD`		
Description	Send command to Flat Device control		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	CommandType	Integer	Command to execute like for relative table
	FlatDeviceIndex	Integer	1 or 2 dependes on which flat device to address
	Brightness	Integer	Only for Set Brightness command type otherwise leave 0
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		

CommandType	Description
1	Open Cover
2	Close Cover
3	Light ON
4	Light OFF
5	Set Brightness

➔ {"method": "RemoteFlatDeviceCMD", "params": {"UID":"e10eacc4-1e60-44d0-bf4a-eab729cf5d5c","FlatDeviceIndex":1,"CommandType":1,"Brightness":0}, "id": 14}

🡐{"jsonrpc": "2.0", "result": 0, "id": 14}

🡐{"Event":"RemoteActionResult","Timestamp":1556990521.31099,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{}}

eee) RemoteRoboDataGetGeoDataCache

<table>
<tr><td>Method</td><td colspan="3">RemoteRoboDataGetGeoDataCache</td></tr>
<tr><td>Description</td><td colspan="3">Send request for info about the location of the mount like stored in Voyager Geo Data Cache</td></tr>
<tr><td>Params</td><td colspan="3"><table><tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr></table></td></tr>
<tr><td>Result</td><td colspan="3">Integer(0)</td></tr>
<tr><td>License Required</td><td colspan="3">Base, Advanced, Full, Custom</td></tr>
<tr><td>CMD result pameters</td><td colspan="3">data object with<table><tr><td>Latitudine</td><td>numeric</td><td>Latitude of the mount location as configured in Voyager (or in mount driver). Null if cache is not configured or empty</td></tr><tr><td>Longitudine</td><td>numeric</td><td>Longitude of the mount as configured in Voyager (or in mount driver). Null if cache is not configured or empty</td></tr><tr><td>Elevation</td><td>numeric</td><td>Elevation in meters of the mount location as configured in Voyager (or in mount driver). Null if cache is not configured or empty</td></tr><tr><td>RemoteDateTime</td><td>Datetime</td><td>LocalTime now</td></tr><tr><td>TimeZoneHour</td><td>numeric</td><td>Time zone expressed in hours from the UTC</td></tr></table></td></tr>
</table>

➔ {"method": "RemoteRoboDataGetGeoDataCache", "params": {"UID":"5f896393-75ad-4ba0-a748-d3d8b7040eb9"}, "id": 12}

🡐{"jsonrpc": "2.0", "result": 0, "id": 12}

🡐{"Event":"RemoteActionResult","Timestamp":1686418111.22173,"Host":"ORIONE","Inst":1,"UID":"5f896393-75ad-4ba0-a748-d3d8b7040eb9","ActionResultInt":4,"Motivo":"","ParamRet":{"data":{
"Latitudine":45.0136111111111, "Longitudine":6.93972222222222, "Elevation":1000,
"RemoteDateTime":1686418111.16039, "TimeZoneHour":2 }}}

fff) RemoteMountStatusGetInfo

Method	`RemoteMountStatusGetInfo`
Description	Send request for info about mount status
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*

CMD result pameters

IsMountConnected	boolean	True if the mount is connected to Voyager, false otherwise. When false then other resul parameters are empty or not valuable excluded the Latitude/Longitude and Elevation
RA	string	Actual RA expressed in string format JNow
DEC	string	Actual DEC expressed in string format JNow
RAJ2000	string	Actual RA expressed in string format J2000
DECJ2000	string	Actual DEC expressed in string format J2000
IsParked	boolean	True if the mount si parked, false if is not parked or mount is not connected to Voyager or mount driver do not report this information
Altitude	string	Actual Altitude expressed in string format
Azimuth	string	Actual Azimuth expressed in string format
Pier	string	Report the pier status as for ASCOM string pierWest for mount pointing before meridian and/or with meridian to do pierEast for mount pointing after meridian and with meridian done
TimeToFlip	string	Time toflip or Hour Angle expressed in HH:MM:SS , negative if the mount is before the meridian crossing
FlipStatus	integer	Status of meridian flip in Voyager logic, see table below
IsTracking	boolean	True if the mount has tracking enable
IsSlewing	boolean	True if the mount is slewing to a target
Latitude	numeric	Latitude of the mount location as configured in Voyager (or in mount driver)
Longitude	numeric	Longitude of the mount as configured in Voyager (or in mount driver)
Elevation	numeric	Elevation in meters of the mount location as configured in Voyager (or in mount driver)

FlipStatus	Description	Note
0	Not needed	Pier is West
1	To do	Meridian flip is necessary, Voyager waiting the right internally status
2	Running	In execution

3	Done	Pier is PierEast
4	Unmanageable	FORK Mount
5	ERROR	Internal error or unknow pier status

➔ {"method": "RemoteMountStatusGetInfo", "params": {"UID":"e10eacc4-1e60-44d0-bf4a-eab729cf5d5c"}, "id": 14}

🡐{"jsonrpc": "2.0", "result": 0, "id": 14}

🡐{"Event":"RemoteActionResult","Timestamp":1556990521.31099,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{"IsMountConnected":true,"RA":"06:07:12","DEC":"90° 00' 00\"","RAJ2000":"23:59:24","DECJ2000":"89° 52' 10\"","IsParked":false,"Altitude":"45° 00' 49\"","Azimuth":"360° 00' 00\"","Pier":"pierWest","TimeToFlip":"-04:18:36","FlipStatus":0,"IsTracking":false,"IsSlewing":false,"Latitude":45.0136111111111,"Longitude":6.93972222222222,"Elevation":1000}}

7. RoboClip Commands

This command are dedicated to RoboClip automata for Targets sharing between Voyager application and modules.

a) RemoteRoboClipGetTargetList

<table>
<tr><td>Method</td><td colspan="3">RemoteRoboClipGetTargetList</td></tr>
<tr><td>Description</td><td colspan="3">RoboClip command. Retrieve the list of Targets in database.</td></tr>
<tr><td>Params</td><td colspan="3">
<table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr>
<tr><td>FilterName</td><td>String</td><td>Optional string to search in target field</td></tr>
<tr><td>FilterGroup</td><td>String</td><td>Optional string to search in group field</td></tr>
<tr><td>FilterNote</td><td>String</td><td>Optional string to search in note field</td></tr>
<tr><td>Order</td><td>Integer</td><td>Target list order to use. See table below</td></tr>
</table>
</td></tr>
<tr><td>Result</td><td colspan="3">Integer(0)</td></tr>
<tr><td>License Required</td><td colspan="3">Base,Advanced, Full, Custom</td></tr>
<tr><td>Remote Action Result Parameters</td><td colspan="3">
<table>
<tr><td>List</td><td>Array</td><td>Array of Target Objects
<table>
<tr><td>guid</td><td>String</td><td>UID of Object</td></tr>
<tr><td>targetname</td><td>String</td><td>Name of Target</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

raj2000	Numeric	Double representing the RA coordinate of target in J2000
decj2000	Numeric	Double representing the DEC coordinate of target in J2000
pa	Numeric	Position angle 0-360°
datecreation	Numeric	Epoch of the date of creation of the target
gruppo	String	Group Name of Target
note	String	Free memo text
ismosaic	Boolean	True if this is a Virtual FOV Voyager Mosaic
frow	Numeric	Number of rows in mosaic
fcol	Numeric	Number of cols in mosaic
tiles	String	CSV file about tiles. Format is TileName;RA;DEC;PA where RA DEC and PA are expressed in double numeric

Order	Description
0	Date of adding to database descending
1	Target Name
2	Group Name + Target Name
3	RA Desc
4	RA Asc

➔ {"method": "RemoteRoboClipGetTargetList", "params": {"FilterName":"","FilterGroup":"","FilterNote":"","Order":0,"UID":"bea5dfcd-c846-4689-a244-a0faea3b3ac1"}, "id": 61}

➔{"jsonrpc": "2.0", "result": 0, "id": 61}

➔ {"Event":"RemoteActionResult","Timestamp":1577048052.93068,"Host":"hal9000","Inst":1,"UID": "bea5dfcd-c846-4689-a244-a0faea3b3ac1","ActionResultInt":4,"Motivo":"","ParamRet":{"list":[{ "guid":"627332c1-be59-4a2d-a277-c1ca83e0fe0c", "targetname":"Zuzzolona", "raj2000":12, "decj2000":13, "pa":0, "datecreation":1577048027, "gruppo":"aloa", "note":"Imported from file RSVoyager.csv" }]}}

b) RemoteRoboClipAddTarget

<table>
<tr><td>Method</td><td>RemoteRoboClipAddTarget</td></tr>
<tr><td>Description</td><td>RoboClip command. Add Target in database.</td></tr>
<tr><td>Params</td><td>
<table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr>
<tr><td>GuidTarget</td><td>String</td><td>UID of the Object</td></tr>
<tr><td>TargetName</td><td>String</td><td>Name of Target</td></tr>
<tr><td>RAJ2000</td><td>Numeric</td><td>Double representing the RA coordinate of target in J2000</td></tr>
<tr><td>DECJ2000</td><td>Numeric</td><td>Double representing the DEC coordinate of target in J2000</td></tr>
<tr><td>PA</td><td>Numeric</td><td>Position angle 0-360°</td></tr>
<tr><td>Group</td><td>String</td><td>Group Name of Target</td></tr>
<tr><td>Note</td><td>String</td><td>Group Name of Target</td></tr>
<tr><td>IsMosaic</td><td>Boolean</td><td>True is is Virtual FOV Voyager Mosaic</td></tr>
<tr><td>FROW</td><td>Numeric</td><td>Number of rows in mosaic</td></tr>
<tr><td>FCOL</td><td>Numeric</td><td>Number of cols in mosaic</td></tr>
<tr><td>TILES</td><td>String</td><td>CSV file about tiles. Format is TileName;RA;DEC;PA where RA DEC and PA are expressed in double numeric</td></tr>
<tr><td>angleAdj</td><td>Boolean</td><td>True if the Mosaic have tiles adjusted for rotation on PA to correct sky pole</td></tr>
<tr><td>overlap</td><td>Numeric</td><td>Overlap value of tiles</td></tr>
<tr><td>DX</td><td>Numeric</td><td>Horizontal count of camera pixels</td></tr>
<tr><td>DY</td><td>Numeric</td><td>Vertical count of camera pixels</td></tr>
<tr><td>PixelSize</td><td>Numeric</td><td>Pixel size of one micron</td></tr>
<tr><td>Focallen</td><td>Numeric</td><td>Focal length express in millimeters</td></tr>
</table>
</td></tr>
<tr><td>Result</td><td>Integer(0)</td></tr>
<tr><td>License Required</td><td>Base, Advanced, Full, Custom</td></tr>
<tr><td>Remote Action Result Parameters</td><td>
<table>
<tr><td>ret</td><td>String</td><td>“DONE” if ok otherwise is an error</td></tr>
</table>
</td></tr>
</table>

➔ {"method": "RemoteRoboClipAddTarget", "params": {"GuidTarget":"AE305703-9453-0A43-0E92-6E2E6E25B406","TargetName":"MOSAIC_TEST","RAJ2000":23.082,"DECJ2000":12.3228,"PA":0,"Group":"MOSAICI","Note":"Prova di Mosaico 13","IsMosaic":true,"FROW":2,"FCOL":2,"TILES":"PANE 1;1.4210;32.2431;0\r\nPANE 2;1.2534;32.2431;0\r\nPANE 3;1.4152;28.5427;0\r\nPANE 4;1.2552;28.5427;0","angleAdj":true,"overlap":1,"DX":2,"DY":3,"PixelSize":4,"Focallen":5,"UID":"603527d1-94d4-4002-bf87-0a6cecbf82bc"}, "id": 13}

←{"jsonrpc": "2.0", "result": 0, "id": 38}

← {"Event":"RemoteActionResult","Timestamp":1577047913.63716,"Host":"hal9000","Inst":1, "UID":"603527d1-94d4-4002-bf87-0a6cecbf82bc ","ActionResultInt":4,"Motivo":"", "ParamRet":{"ret":"DONE"}}

c) RemoteRoboClipRemoveTarget

Method	RemoteRoboClipRemoveTarget		
Description	RoboClip command. Remove Target in database.		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	RefGuidTarget	String	UID of the Object [empty to remove all the targets !!!!]
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		
Remote Action Result Parameters	ret	String	"DONE" if ok otherwise is an error

→ {"method": "RemoteRoboClipRemoveTarget", "params": {"RefGuidTarget":"564e9ef8-f190-4e74-84ab-b9d651c48531","UID":"bc8ca246-b678-4cc3-a0b0-18eb56c12f77"}, "id": 9}

←{"jsonrpc": "2.0", "result": 0, "id": 38}

← {"Event":"RemoteActionResult","Timestamp":1577034908.31403,"Host":"hal9000","Inst":1, "UID":"bc8ca246-b678-4cc3-a0b0-18eb56c12f77","ActionResultInt":4,"Motivo":"", "ParamRet":{"ret":"DONE"}}

d) RemoteRoboClipUpdateTarget

Method	RemoteRoboClipUpdateTarget		
Description	RoboClip command. Update Target in database.		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	RefGuidTarget	String	UID of the Object

	TargetName	String	Name of Target
	RAJ2000	Numeric	Double representing the RA coordinate of target in J2000
	DECJ2000	Numeric	Double representing the DEC coordinate of target in J2000
	PA	Numeric	Position angle 0-360°
	Group	String	Group Name of Target
	Note	String	Group Name of Target
	IsMosaic	Boolean	True if is Virtual FOV Voyager Mosaic
	FROW	Numeric	Number of rows in mosaic
	FCOL	Numeric	Number of cols in mosaic
	TILES	String	CSV file about tiles. Format is TileName;RA;DEC;PA where RA DEC and PA are expressed in double numeric
	angleAdj	Boolean	True if the Mosaic have tiles adjusted for rotation on PA to correct sky pole
	overlap	Numeric	Overlap value of tiles
	DX	Numeric	Horizontal count of camera pixels
	DY	Numeric	Vertical count of camera pixels
	PixelSize	Numeric	Pixel size of one micron
	Focallen	Numeric	Focal length express in millimeters
Result	Integer(0)		
License Required	*Base, Advanced, Full, Custom*		
Remote Action Result Parameters	ret	String	“DONE” if ok otherwise is an error

➔ {"method": "RemoteRoboClipUpdateTarget", "params": {"RefGuidTarget":"FE305703-9453-0A43-0E92-6E2E6E25B406","TargetName":"MOSAIC_TEST","RAJ2000":23.082,"DECJ2000":12.3228,"PA":0,"Group":"MOSAICI","Note":"Prova di Mosaico 11","IsMosaic":true,"FROW":2,"FCOL":2,"TILES":"PANE 1;1.4210;32.2431;0\r\nPANE 2;1.2534;32.2431;0\r\nPANE 3;1.4152;28.5427;0\r\nPANE 4;1.2552;28.5427;0","angleAdj":false,"overlap":0,"DX":0,"DY":0,"PixelSize":0,"Focallen":0,"UID":"58292091-9c14-4d85-8b3b-cd86bc837b50"}, "id": 9}

➔{"jsonrpc": "2.0", "result": 0, "id":11}

➔ {"Event":"RemoteActionResult","Timestamp":1577041554.11737,"Host":"hal9000","Inst":1, "UID":"58292091-9c14-4d85-8b3b-cd86bc837b50","ActionResultInt":4,"Motivo":"", "ParamRet":{"ret":"DONE"}}

8. RoboTarget Commands

This commands are dedicated to RoboTarget automata, are only available starting from the Advanced version of Voyager. Here are listed only the commands open to all types of client of the Application Server and free to use. For all other commands exposed by this very powerful Automa, a dedicated NDA is required, contact the Voyager sales team at voyagerastro@gmail.com)

All the Open Commands of RoboTarget need a MAC validation, you must cknow the RoboTarget shared secret of remote Application Server to compose the MAC.

If you want to rating the shot done by Voyager RoboTarget automation:

1) Retrieve the list of target with `RemoteOpenRoboTargetGetTargetList`
2) Select the interested Target GUID
3) Ask the Shot Done for the target with GUID selected at point 2 using `RemoteOpenRoboTargetGetShotDoneList`
4) Select the interested Shot Done GUID
5) Set the Rating using the GUID selected at point 2 using `RemoteOpenRoboTargetSetShotDoneRating`

GENERAL CONCEPT ON DELETED SHOT DONE in Voyager RoboTarget:

All the Open RoboTarget Command cannot delete physically a file on HDD. The delete operations are only logical operation that set or unset a boolean flag on data records. So you can delete and restore all the shot done at any time without touch the physical FIT file. Just the Deleted Shots done are not used to calculate progress. So if you delete a shot done for a finished target this will means to the scheduler to take back in the game the target to finish it according the new progress.

a) RemoteOpenRoboTargetGetTargetList

<table>
<tr><td>Method</td><td colspan="3"><code>RemoteOpenRoboTargetGetTargetList</code></td></tr>
<tr><td>Description</td><td colspan="3">RoboTarget command. Retrieve the list of all the Target configured in RoboTarget Automata database.</td></tr>
<tr><td>Params</td><td colspan="3"><table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr>
<tr><td>MAC</td><td>String</td><td>Create a concatenated string with RoboTarget Shared secret + UID of the action (previous parameter) and make an MD5 hash, see the example below.</td></tr>
</table></td></tr>
<tr><td>Result</td><td colspan="3">Integer(0)</td></tr>
<tr><td>License Required</td><td colspan="3">Advanced, Full</td></tr>
</table>

Remote Action Result Parameters

List	Array	Array of Target Objects

guid	String	UID of Object
targetname	String	Name of Target
tag	string	Tag of Target
datecreation	Numeric	Epoch of the date of creation of the target
status	Numeric	Status of Target ENABLED = 0 DISABLED = 1
statusop	Numeric	Operative Status of Target UNKNOW = -1 IDLE = 0 RUNNING = 1 FINISHED = 2 NO_EPHEM = 3
setname	String	Name of Set contains the Target
settag	string	Tag of Set
profilename	String	Name of Voyager Setup Profile file where is associated the Target (with file extension)

➔ {"method": "RemoteOpenRoboTargetGetTargetList", "params": {"UID":"d4a644d7-10d2-4904-9de4-9c1ec5cf6a6a","MAC":"684660d3045ee9c2bbc626a4e5cc5155"}, "id": 37}

←{"jsonrpc": "2.0", "result": 0, "id":37}

←

{"Event":"RemoteActionResult","Timestamp":1647175864.14868,"Host":"ORIONE","Inst":1,"UID":"d4a644d7-10d2-4904-9de4-9c1ec5cf6a6a","ActionResultInt":4,"Motivo":"","ParamRet":{"list":[{"guid":"2d155808-ee20-4036-b595-8002330be5a0","targetname":"Birillo","datecreation":1644848898,"status":0,"statusop":0,"setname":"Test","profilename":"TestFlatNoMount.v2y"},{"guid":"4a8a9d40-759c-414f-b62a-8a633f4d3cf1","targetname":"Bubble Nebula","datecreation":1639854721,"status":0,"statusop":2,"setname":"Finished","profilename":"Default.v2y"},{"guid":"4f15fe4f-2970-43a2-9aa5-98c1000bd6a3","targetname":"Cone Nebula","datecreation":1642031291,"status":0,"statusop":0,"setname":"Narrow HAOIII","profilename":"Default.v2y"},{"guid":"33e96ef9-8d6f-4f19-b443-

30e3285ac3cc","targetname":"Doppietto Leone","datecreation":1643502804,"status":0,"statusop":0,"setname":"Galaxy","profilename":"Default.v2y]}}

MAC creation for this call with a RoboTarget Shared Secret = leonardo

String concatenated = leonardod4a644d7-10d2-4904-9de4-9c1ec5cf6a6a

MAC with MD5 hashing = 684660d3045ee9c2bbc626a4e5cc5155

You can check also creating with online tools for MD5 hashing

b) RemoteOpenRoboTargetGetShotDoneList

<table>
<tr><td>Method</td><td>RemoteOpenRoboTargetGetShotDoneList</td></tr>
<tr><td>Description</td><td>RoboTarget command. Retrieve the list of all the Shot Done for the requested Target UID from the database. Not necessary the file exists more in the physical disk of remote PC.</td></tr>
<tr><td>Params</td><td>
<table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr>
<tr><td>RefGuidTarget</td><td>String</td><td>Unique identifier of the Target to use for retrieve all the Shot Done and registered in RoboTarget database (get it from the Target list obtained with the previous command RemoteOpenRoboTargetGetTargetList)</td></tr>
<tr><td>IsDeleted</td><td>Bool</td><td>True to obtain the list of Shot Done and Deleted (for a rating or a manual user decision), False to obtain the list of Shot Done and not deleted. Remember that deleted Shot Done is just a logical flag on data record. Deleted Shot done are not used to calculate the target progress.</td></tr>
<tr><td>MAC</td><td>String</td><td>Create a concatenated string with RoboTarget Shared secret + UID of the action (previous parameter) + RefGuidTarget and make an MD5 hash, see the example below.</td></tr>
</table>
</td></tr>
<tr><td>Result</td><td>Integer(0)</td></tr>
<tr><td>License Required</td><td>Advanced, Full</td></tr>
<tr><td>Remote Action Result Parameters</td><td>
<table>
<tr><td>List</td><td>Array</td><td>Array of Target Objects
<table>
<tr><td>guid</td><td>String</td><td>UID of Object</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

			datetimeshot	Datetime	Date time when shot was done
			datetimeshotutc	Datetime	Date time UTC when shot was done
			filename	String	File name of FIT
			hfd	Numeric	Half Flux Diameter of stars on the Image (average of all stars in the field)
			max	Numeric	Max ADU value of the image
			mean	Numeric	Average ADU value of the image
			min	Numeric	Min ADU value of the image
			path	String	Path of the FIT file if available
			refguidsession	String	Unique Identifier of the Session where the shot was done
			refguidshot	String	Unique Identifier of the Shot configuration used
			starindex	Numeric	Decimal value indicating the star presence in the image
			bin	Numeric	Binning used for the shot
			filterindex	Numeric	Filter index used for the shot
			exposure	Numeric	Exposure express in seconds for shot
			rating	Numeric	External rating integer value indicating the quality of FIT. <=0 = not evaluated >0 evaluated. More is high better is the image quality. This rating value is not provided by Voyager, you must user external tools capable to link to Voyager RoboTarget or you can create your own tool

			isdeleted	bool	Indicate if the shot done is logically deleted by user

➔ {"method": "RemoteOpenRoboTargetGetShotDoneList", "params": {"RefGuidTarget":"6c5553ef-3c11-4b40-a3e1-7cd008e08c35","IsDeleted":false, "UID":"d4a644d7-10d2-4904-9de4-9c1ec5cf6a6a","MAC":"0241332cd7da9ec94e5a839fcee41ab4"}, "id": 37}

➔{"jsonrpc": "2.0", "result": 0, "id":37}

➔
{"Event":"RemoteActionResult","Timestamp":1647176657.5084,"Host":"ORIONE","Inst":1,"UID":"d4a644d7-10d2-4904-9de4-9c1ec5cf6a6a","ActionResultInt":4,"Motivo":"","ParamRet":{"list":[{"guid":"120e5c72-9aae-4363-8fc4-f0105aa4c4b3","datetimeshot":1640715509,"filename":"M31_LRGB_LIGHT_L_300s_BIN1_-12C_001_20211228_181829_437_GA_1087_OF_60_W.FIT","hfd":6.45,"max":65535,"mean":18649,"min":0,"path":"","refguidsession":"cf996602-8e6b-4461-8cbf-81d813e9893f","refguidshot":"73cead8d-4f75-4a15-8db3-bea3d0281343","starindex":20.65,"bin":1,"filterindex":0,"exposure":300,"rating":14,"isdeleted":false}]}}

MAC creation for this call with a RoboTarget Shared Secret = leonardo

String concatenated = leonardod4a644d7-10d2-4904-9de4-9c1ec5cf6a6a6c5553ef-3c11-4b40-a3e1-7cd008e08c35

MAC with MD5 hashing = 0241332cd7da9ec94e5a839fcee41ab4

You can check also creating with online tools for MD5 hashing

c) RemoteOpenRoboTargetSetShotDoneRating

Method	RemoteOpenRoboTargetSetShotDoneRating		
Description	RoboTarget command. Set the Rating value of a Shot Done .		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	ObjUID	String	UID of the Shot Done / Session / Target (you must use the GUID reported by the previous commands) RemoteOpenRoboTargetGetShotDoneList
	Mode	Numeric	Define how to work with the ObjUID to affect one or more Shot Done (for example by Session

			all the Shot Done for the selected Session will be updated with the same rating value). By Shot = 0 ; By Session = 1 ; By Target=2; By Slot = 3
	Rating	Numeric	Integer value indicating the quality of FIT. <=0 : not evaluated >0 : evaluated. More is high better is the image quality.
	IsDeleted	Bool	True to Apply the command ONLY to the Deleted Shot Done and restore it if needed , False to Apply the command ONLY to the NOT Deleted Shot Done and delete if needed
	MAC		Create a concatenated string with RoboTarget Shared secret + UID of the action (previous parameter) + RefGuidShotDone and make an MD5 hash, see the example below.
Result	Integer(0)		
License Required	*Advanced, Full*		
Remote Action Result Parameters	ret	String	"DONE" if ok otherwise is an error

➔ {"method": "RemoteOpenRoboTargetSetShotDoneRating", "params": {"ObjUID":"120e5c72-9aae-4363-8fc4-f0105aa4c4b3","Mode":0,"Rating":14,"IsDeleted":false,"UID":"d4a644d7-10d2-4904-9de4-9c1ec5cf6a6a" ,"MAC":"644a08429b66cecfafa4d0251f576639" }, "id": 37}

←{"jsonrpc": "2.0", "result": 0, "id":37}

←

{"Event":"RemoteActionResult","Timestamp":1647177587.31445,"Host":"ORIONE","Inst":1,"UID":"d4a644d7-10d2-4904-9de4-9c1ec5cf6a6a","ActionResultInt":4,"Motivo":"","ParamRet":{"ret":"DONE"}}

MAC creation for this call with a RoboTarget Shared Secret = leonardo

String concatenated = leonardod4a644d7-10d2-4904-9de4-120e5c72-9aae-4363-8fc4-f0105aa4c4b3

MAC with MD5 hashing = 644a08429b66cecfafa4d0251f576639

You can check also creating with online tools for MD5 hashing

d) RemoteOpenRoboTargetRemoveShotDone

<table>
<tr><td>Method</td><td>RemoteOpenRoboTargetRemoveShotDone</td></tr>
<tr><td>Description</td><td>RoboClip command. Remove Shot Done From Database. Do not remove the file from the Voyager PC.</td></tr>
<tr><td>Params</td><td>
<table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr>
<tr><td>ObjUID</td><td>String</td><td>UID of the Shot Done / Session / Target / Set (you must use the GUID reported by the previous commands) RemoteOpenRoboTargetGetShotDoneList</td></tr>
<tr><td>Mode</td><td>Numeric</td><td>Define how to work with the ObjUID to affect one or more Shot Done (for example by Session all the Shot Done for the selected Session will be delete).
By Shot = 0 ; By Session = 1 ; By Target=2; By Slot =3 ; By Set = 4</td></tr>
<tr><td>RatingMode</td><td>Numeric</td><td>Define if delete by Rating or Not
None = 0 (delete all) ; Lower Limit = 1 (only Shot Done with Rating < RatingLimit) ; Greater Limit = 2 (only Shot Done with Rating > RatingLimit)</td></tr>
<tr><td>RatingLimit</td><td>Numeric</td><td>Integer value indicating the quality of FIT</td></tr>
<tr><td>MAC</td><td></td><td>Create a concatenated string with RoboTarget Shared secret + UID of the action (previous parameter) + RefGuidShotDone and make an MD5 hash, see the example below.</td></tr>
</table>
</td></tr>
<tr><td>Result</td><td>Integer(0)</td></tr>
<tr><td>License Required</td><td>Advanced, Full</td></tr>
<tr><td>Remote Action Result Parameters</td><td>
<table>
<tr><td>ret</td><td>String</td><td>“DONE” if ok otherwise is an error</td></tr>
</table>
</td></tr>
</table>

➔ {"method": "RemoteOpenRoboTargetRemoveShotDone", "params": {" ObjUID ":"120e5c72-9aae-4363-8fc4-f0105aa4c4b3”, "UID":"d4a644d7-10d2-4904-9de4-9c1ec5cf6a6a" ,"Mode":0, "RatingMode":1, "RatingLimit": 7,"MAC":"644a08429b66cecfafa4d0251f576639" }, "id": 37}

←{"jsonrpc": "2.0", "result": 0, "id":37}

←

{"Event":"RemoteActionResult","Timestamp":1647177587.31445,"Host":"ORIONE","Inst":1,"UID":"d4a644d7-10d2-4904-9de4-9c1ec5cf6a6a","ActionResultInt":4,"Motivo":"","ParamRet":{"ret":"DONE"}}

MAC creation for this call with a RoboTarget Shared Secret = leonardo

String concatenated = leonardod4a644d7-10d2-4904-9de4-120e5c72-9aae-4363-8fc4-f0105aa4c4b3

MAC with MD5 hashing = 644a08429b66cecfafa4d0251f576639

You can check also creating with online tools for MD5 hashing

e) RemoteOpenRoboTargetRestoreShotDone

<table>
<tr><td>Method</td><td>RemoteOpenRoboTargetRestoreShotDone</td></tr>
<tr><td>Description</td><td>RoboClip command. Restore Shot Done From Database.</td></tr>
<tr><td>Params</td><td>
<table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr>
<tr><td>ObjUID</td><td>String</td><td>UID of the Shot Done / Session / Target / Set (you must use the GUID reported by the previous commands) RemoteOpenRoboTargetGetShotDoneList</td></tr>
<tr><td>Mode</td><td>Numeric</td><td>Define how to work with the ObjUID to affect one or more Shot Done (for example by Session all the Shot Done for the selected Session will be restore).
By Shot = 0 ; By Session = 1 ; By Target=2; By Slot=3; By Set = 4</td></tr>
<tr><td>RatingMode</td><td>Numeric</td><td>Define if restore by Rating or Not
None = 0 (restore all) ; Lower Limit = 1 (only Shot Done with Rating < RatingLimit) ; Greater Limit = 2 (only Shot Done with Rating > RatingLimit)</td></tr>
<tr><td>RatingLimit</td><td>Numeric</td><td>Integer value indicating the quality of FIT</td></tr>
<tr><td>MAC</td><td></td><td>Create a concatenated string with RoboTarget Shared secret + UID of the action (previous parameter) + RefGuidShotDone and make an MD5 hash, see the example below.</td></tr>
</table>
</td></tr>
<tr><td>Result</td><td>Integer(0)</td></tr>
<tr><td>License Required</td><td>Advanced, Full</td></tr>
<tr><td>Remote Action Result Parameters</td><td>
<table>
<tr><td>ret</td><td>String</td><td>“DONE” if ok otherwise is an error</td></tr>
</table>
</td></tr>
</table>

➔ {"method": "RemoteOpenRoboTargetRestoreShotDone", "params": {" ObjUID ":"120e5c72-9aae-4363-8fc4-f0105aa4c4b3", "UID":"d4a644d7-10d2-4904-9de4-9c1ec5cf6a6a" ,"Mode":0, "RatingMode":1, "RatingLimit": 7,"MAC":"644a08429b66cecfafa4d0251f576639" }, "id": 37}

➔{"jsonrpc": "2.0", "result": 0, "id":37}

➔
{"Event":"RemoteActionResult","Timestamp":1647177587.31445,"Host":"ORIONE","Inst":1,"UID":"d4a644d7-10d2-4904-9de4-9c1ec5cf6a6a","ActionResultInt":4,"Motivo":"","ParamRet":{"ret":"DONE"}}

MAC creation for this call with a RoboTarget Shared Secret = leonardo

String concatenated = leonardod4a644d7-10d2-4904-9de4-120e5c72-9aae-4363-8fc4-f0105aa4c4b3

MAC with MD5 hashing = 644a08429b66cecfafa4d0251f576639

You can check also creating with online tools for MD5 hashing

f) RemoteOpenRoboTargetUpdateBulkShotDone

<table>
<tr><td>Method</td><td colspan="3">RemoteOpenRoboTargetUpdateBulkShotDone</td></tr>
<tr><td>Description</td><td colspan="3">RoboTarget command. Bulk update of Rating and/or Delete of shot done from an array of objects</td></tr>
<tr><td>Params</td><td colspan="3">
<table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr>
<tr><td>SrcList</td><td>Array of Object</td><td>Array of Shot Done Object
<table>
<tr><td>RefGuidShotDone</td><td>String</td><td>UID of Shot Done like retrieved by RoboTarget Open GetShotDoneList</td></tr>
<tr><td>Rating</td><td>Numeric</td><td>Integer value indicating the quality of FIT.
<=0 : not evaluated
>0 : evaluated.

More is high better is the image quality.</td></tr>
<tr><td>IsToDelete</td><td>Boolean</td><td>True if you want to delete the Shot Done</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

	IsDeleted	Bool	True to Apply the command ONLY to the Deleted Shot Done and restore it if needed , False to Apply the command ONLY to the NOT Deleted Shot Done and delete if needed
	MAC		Create a concatenated string with RoboTarget Shared secret + UID of the action (previous parameter) and make an MD5 hash, see the example below.
Result	Integer(0)		
License Required	*Advanced, Full*		
Remote Action Result Parameters	ret	String	“DONE” if ok otherwise is an error

➔ {"method": "RemoteOpenRoboTargetUpdateBulkShotDone", "params": {"SrcList":[{"RefGuidShotDone":"13c32f52-1649-4184-82fe-3eebb25005d5","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"271a053e-04e7-4747-b1b1-b0ab20351c55","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"5216350b-f695-49ff-bf4e-2df9cff01ad2","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"cf358714-b5ec-4250-837b-294a459cc5e9","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"4329cfee-36d4-489e-9466-0ce07b257524","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"642455a9-64b5-461e-a389-f56e2d207a28","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"4841886e-5606-4e6f-9702-54e3885badda","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"6d5001ab-514b-48cc-ac7e-b98de318488b","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"85cea743-cc7c-4e05-b59e-0d1781e1613a","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"90e463a6-bf13-4942-8d9e-1eabee980c2d","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"9a4dc598-51af-4a3b-8428-1e929bcd591c","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"f1ccf07f-df07-467e-8c9a-ab56dd6ce0bd","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"53469c88-a37b-4864-b75b-9f0d8f56c4bd","Rating":233,"IsToDelete":false},{"RefGuidShotDone":"ae92ccca-9916-4fe5-8fdf-16dbfc4843d8","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"50ae14a7-1eda-495b-99b6-b1cc90fa72b4","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"04130464-8ad9-47af-98e7-a7752383aa3d","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"3e2ff89c-5114-4be2-955c-cb67d726fd15","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"9f75ecfb-25a6-4b31-83a1-8be20cd121ac","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"eabe4019-6bb4-4fc6-bd46-86e8988be1fa","Rating":0,"IsToDelete":false},{"RefGuidShotDone":"be0341c3-c382-4cc3-b8c8-a8306b68a54a","Rating":0,"IsToDelete":false}],"IsDeleted":false,"UID":"88588e1b-bd6e-4008-a27e-9c0be2abd242","MAC":"5f98c3681a26bb2c1415e3342d46014c"}, "id": 31}

➔{"jsonrpc": "2.0", "result": 0, "id":31}

➔

{"Event":"RemoteActionResult","Timestamp":1647177587.31445,"Host":"ORIONE","Inst":1,"UID":"88588e1b-bd6e-4008-a27e-9c0be2abd242","ActionResultInt":4,"Motivo":"","ParamRet":{"ret":"DONE"}}

MAC creation for this call with a RoboTarget Shared Secret = leonardo

String concatenated = leonardod4a644d7-10d2-4904-9de4-120e5c72-9aae-4363-8fc4-f0105aa4c4b3

MAC with MD5 hashing = 644a08429b66cecfafa4d0251f576639

You can check also creating with online tools for MD5 hashing

g) RemoteOpenRoboTargetRemoveShotDoneByFileName

<table>
<tr><td>Method</td><td>RemoteOpenRoboTargetRemoveShotDoneByFileName</td></tr>
<tr><td>Description</td><td>RoboClip command. Remove Shot Done From Database. Do not remove the file from the Voyager PC.</td></tr>
<tr><td>Params</td><td>
<table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr>
<tr><td>FileNameFIT</td><td>String</td><td>Name of the fit file to remove with extension</td></tr>
<tr><td>MAC</td><td></td><td>Create a concatenated string with RoboTarget Shared secret + UID of the action (previous parameter) + FileNameFIT and make an MD5 hash, see the example below.</td></tr>
</table>
</td></tr>
<tr><td>Result</td><td>Integer(0)</td></tr>
<tr><td>License Required</td><td>Advanced, Full</td></tr>
<tr><td>Remote Action Result Parameters</td><td>
<table>
<tr><td>ret</td><td>String</td><td>“DONE” if ok otherwise is an error</td></tr>
</table>
</td></tr>
</table>

➔ {"method": "RemoteOpenRoboTargetRemoveShotDoneByFileName", "params": {"
FileNameFIT":"M31_LRGB_LIGHT_L_300s_BIN1_-
12C_003_20211228_182925_847_GA_1087_OF_60_W.FIT","UID":"d4a644d7-10d2-4904-9de4-
9c1ec5cf6a6a","MAC":"f6b655c1e990f321c1b2238efe70a971"}, "id": 37}

🡐{"jsonrpc": "2.0", "result": 0, "id":37}

🡐
{"Event":"RemoteActionResult","Timestamp":1647177587.31445,"Host":"ORIONE","Inst":1,"UID":"d4a644
d7-10d2-4904-9de4-9c1ec5cf6a6a","ActionResultInt":4,"Motivo":"","ParamRet":{"ret":"DONE"}}

MAC creation for this call with a RoboTarget Shared Secret = leonardo

String concatenated = leonardod4a644d7-10d2-4904-9de4-120e5c72-9aae-4363-8fc4-f0105aa4c4b3 M31_LRGB_LIGHT_L_300s_BIN1_-12C_003_20211228_182925_847_GA_1087_OF_60_W.FIT

MAC with MD5 hashing = f6b655c1e990f321c1b2238efe70a971

You can check also creating with online tools for MD5 hashing

h) RemoteOpenRoboTargetRestoreShotDoneByFileName

Method	RemoteOpenRoboTargetRestoreShotDoneByFileName		
Description	RoboClip command. Restore Shot Done From Database. The file must be logically deleted by a previous user action.		
Params	UID	String	Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated
	FileNameFIT	String	Name of the fit file to restore with extension
	MAC		Create a concatenated string with RoboTarget Shared secret + UID of the action (previous parameter) + FileNameFIT and make an MD5 hash, see the example below.
Result	Integer(0)		
License Required	*Advanced, Full*		
Remote Action Result Parameters	ret	String	"DONE" if ok otherwise is an error

➔ {"method": "RemoteOpenRoboTargetRestoreShotDoneByFileName", "params": {"FileNameFIT":"M31_LRGB_LIGHT_L_300s_BIN1_-12C_003_20211228_182925_847_GA_1087_OF_60_W.FIT","UID":"d4a644d7-10d2-4904-9de4-9c1ec5cf6a6a","MAC":"f6b655c1e990f321c1b2238efe70a971"}, "id": 37}

←{"jsonrpc": "2.0", "result": 0, "id":37}

←

{"Event":"RemoteActionResult","Timestamp":1647177587.31445,"Host":"ORIONE","Inst":1,"UID":"d4a644d7-10d2-4904-9de4-9c1ec5cf6a6a","ActionResultInt":4,"Motivo":"","ParamRet":{"ret":"DONE"}}

MAC creation for this call with a RoboTarget Shared Secret = leonardo

String concatenated = leonardod4a644d7-10d2-4904-9de4-120e5c72-9aae-4363-8fc4-f0105aa4c4b3 M31_LRGB_LIGHT_L_300s_BIN1_-12C_003_20211228_182925_847_GA_1087_OF_60_W.FIT

MAC with MD5 hashing = f6b655c1e990f321c1b2238efe70a971

You can check also creating with online tools for MD5 hashing

i) RemoteOpenRoboTargetSetShotDoneRatingByFileName

Method	RemoteOpenRoboTargetSetShotDoneRatingByFileName
Description	RoboClip command. Upate Rating Shot Done From Database.
Params	UID — String — Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated FileNameFIT — String — Name of the fit file where to update the rating MAC — Create a concatenated string with RoboTarget Shared secret + UID of the action (previous parameter) + FileNameFIT and make an MD5 hash, see the example below.
Result	Integer(0)
License Required	*Advanced, Full*
Remote Action Result Parameters	ret — String — "DONE" if ok otherwise is an error

➔ {"method": "RemoteOpenRoboTargetSetShotDoneRatingByFileName", "params": {"
FileNameFIT":"M31_LRGB_LIGHT_L_300s_BIN1_-
12C_003_20211228_182925_847_GA_1087_OF_60_W.FIT","UID":"d4a644d7-10d2-4904-9de4-
9c1ec5cf6a6a","MAC":"f6b655c1e990f321c1b2238efe70a971"}, "id": 37}

➔{"jsonrpc": "2.0", "result": 0, "id":37}

➔
{"Event":"RemoteActionResult","Timestamp":1647177587.31445,"Host":"ORIONE","Inst":1,"UID":"d4a644
d7-10d2-4904-9de4-9c1ec5cf6a6a","ActionResultInt":4,"Motivo":"","ParamRet":{"ret":"DONE"}}

MAC creation for this call with a RoboTarget Shared Secret = leonardo

String concatenated = leonardod4a644d7-10d2-4904-9de4-120e5c72-9aae-4363-8fc4-f0105aa4c4b3
M31_LRGB_LIGHT_L_300s_BIN1_-12C_003_20211228_182925_847_GA_1087_OF_60_W.FIT

MAC with MD5 hashing = f6b655c1e990f321c1b2238efe70a971

You can check also creating with online tools for MD5 hashing

9. RoboOrbits

This commands are dedicated to RoboOrbits automata for find Comets and Asteroid targets in Voyager Database. You must configure RoboOrbits and import data from Internet source.

s) RemoteRoboOrbitsGetAsteroids

<table>
<tr><td>Method</td><td>RemoteRoboOrbitsGetAsteroids</td></tr>
<tr><td>Description</td><td>Search for asteroids object match the search string in various mode</td></tr>
<tr><td>Params</td><td>
<table>
<tr><td>UID</td><td>String</td><td>Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated</td></tr>
<tr><td>SearchText</td><td>String</td><td>Text to search in asteroids key or designation name (fast search)</td></tr>
<tr><td>SearchType</td><td>Integer</td><td>Search mode, 0=start with the SearchText , 1 = contains the SearchText (deep search</td></tr>
</table>
</td></tr>
<tr><td>Result</td><td>Integer(0)</td></tr>
<tr><td>License Required</td><td>Base, Advanced, Full, Custom</td></tr>
<tr><td>Remote Action Result Parameters</td><td>
<table>
<tr><td>List</td><td>Array</td><td>Array of Asteroid Objects
<table>
<tr><td>MPName</td><td>string</td><td>Key of Asteroid Object like stored in Voyager RoboOrbits Database</td></tr>
<tr><td>AsteroidNumber</td><td>string</td><td>Asteroid Number (is a text can be alphanumeric) if available like imported from Internet source format (Lowell ASTORB)</td></tr>
<tr><td>NamePreliminaryDesignation</td><td>string</td><td>Designation Name for the asteroid like imported from Internet source format (Lowell ASTORB)</td></tr>
<tr><td>RawDataLine</td><td>String</td><td>Full row text line data like imported from Internet source format (Lowell ASTORB)</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

➔ {"method": "RemoteRoboOrbitsGetAsteroids", "params": {"SearchText":"anton","SearchType":1,"UID":"46bd4790-8ea6-44e7-8a12-6a62a968c53c"}, "id": 16}

🡐{"jsonrpc": "2.0", "result": 0, "id": 16}

🡐{"Event":"RemoteActionResult","Timestamp":1686417601.21612,"Host":"ORIONE","Inst":1,"UID":"46bd4790-8ea6-44e7-8a12-6a62a968c53c","ActionResultInt":4,"Motivo":"","ParamRet":{"list":[{ "MPName":"Ahantonioli", "AsteroidNumber":"17984", "NamePreliminaryDesignation":"Ahantonioli", "RawDataLine":" 17984 Ahantonioli L.H. Wasserman 14.37 0.15 0 0 0 0 0 0 9585 2302 20230605 157.124836 61.814261 178.060785 3.150116 0.14382759 2.69772116 20230429 1.1E-02 1.6E-05 20230528 2.3E-02 20231208 4.0E-02 20300531 4.0E-02 20300531\n" },{ "MPName":"Antonacci", "AsteroidNumber":"84120", "NamePreliminaryDesignation":"Antonacci", "RawDataLine":" 84120 Antonacci L.H. Wasserman 15.72 0.15 0 0 0 0 0 0 11395 888 20230605 259.784976 223.601589 179.659915 13.763967 0.18571708 2.66278452 20221125 3.6E-02 1.9E-04 20230528 4.2E-02 20230717 6.3E-02 20281014 6.3E-02 20281014\n" },{ "MPName":"Antonalexander", "AsteroidNumber":"300334", "NamePreliminaryDesignation":"Antonalexander", "RawDataLine":"300334 Antonalexander L.H. Wasserman 16.53 0.15 0 0 0 0 0 0 7572 556 20230605 97.309976 8.375342 277.357795 8.241589 0.24121701 2.42064607 20230429 2.2E-02 3.5E-05 20230528 4.6E-02 20231215 6.0E-02 20290406 8.0E-02 20330515\n"}]}}

t) RemoteRoboOrbitsGetComets

Method	`RemoteRoboOrbitsGetComets`
Description	Search for comets object match the search string
Params	UID (String) — Unique identifier of the Action to abort. Use a Guide Window identifier or a unique key string generated SearchText (String) — Text to search in asteroids key or designation name. Text contains search mode will used.
Result	Integer(0)
License Required	*Base, Advanced, Full, Custom*
Remote Action Result Parameters	List (Array) — Array of Comet Objects CTName (string) — Key of Comet Object like stored in Voyager RoboOrbits Database

PeriodicCometNumber	integer	Comet number like imported from Internet source format (MinorPlanetCenter MPCORB)
DesignationAndName	string	Designation Name for the Comet like imported from Internet source format (MinorPlanetCenter MPCORB)
RawDataLine	String	Full row text line data like imported from Internet source format (MinorPlanetCenter MPCORB)

➔ {"method": "RemoteRoboOrbitsGetCometsLike", "params": {"SearchText":"97","UID":"a38d9d8f-db6e-46f1-b59d-918e44c8d5f3"}, "id": 16}

➔{"jsonrpc": "2.0", "result": 0, "id": 16}

➔{"Event":"RemoteActionResult","Timestamp":1686418131.45737,"Host":"ORIONE","Inst":1,"UID":"a38d9d8f-db6e-46f1-b59d-918e44c8d5f3","ActionResultInt":4,"Motivo":"","ParamRet":{"list":[{ "CTName":"0097P", "DesignationAndName":"97P/Metcalf-Brewington", "PeriodicCometNumber":97, "RawDataLine":"0097P 2022 02 15.7734 2.570997 0.460257 230.0340 184.0778 17.9517 20230603 5.5 6.0 97P/Metcalf-Brewington MPEC 2022-C56" },{ "CTName":"0197P", "DesignationAndName":"197P/LINEAR", "PeriodicCometNumber":197, "RawDataLine":"0197P 2022 12 7.7405 1.063100 0.629402 188.7715 66.3556 25.5290 20230603 16.5 2.0 197P/LINEAR MPC 94676" },{ "CTName":"0297P", "DesignationAndName":"297P/Beshore", "PeriodicCometNumber":297, "RawDataLine":"0297P 2021 01 29.8350 2.357599 0.321013 136.1195 95.5574 10.3357 20230603 15.0 4.0 297P/Beshore MPC102956" },{ "CTName":"0397P", "DesignationAndName":"397P/Lemmon", "PeriodicCometNumber":397, "RawDataLine":"0397P 2020 06 19.9546 2.278150 0.405248 14.5449 8.1955 10.9288 20230603 12.0 4.0 397P/Lemmon MPEC 2022-OB6" }]}}

10. Donuts Management

This commands & events are dedicated to DONUTS centering software.

u) RemoteSetDonutsMods

Method	`RemoteSetDonutsMode`
Description	When the client connect to Application Server can specify if is a Donuts. If a client is a Donuts Client , the Application Server will send a `ControlData` event
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*

➔ {"method": "RemoteSetDonutsMode", "params": {"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8" }, "id": 2}

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

←{"Event":"RemoteActionResult","Timestamp":1556990521.31099,"Host":"hal9000","Inst":1,"UID":"eaea5429-f5a9-4012-bc9b-f109e605f5d8","ActionResultInt":4,"Motivo":"","ParamRet":{}}

v) DonutsAbort

Raised when Voyager Ask to DONUTS external application to Abort Actual Operation.

Example:

{"Event":"DonutsAbort","Timestamp":1619784510.33227,"Host":"ORIONE","Inst":1}

w) DonutsCalibrationRequired

Raised when Voyager Ask to DONUTS external application to start the Calibration Task.

Example:

{"Event":"DonutsCalibrationRequired","Timestamp":1619784510.33227,"Host":"ORIONE","Inst":1}

x) DonutsCalibrationStart

Raised when DONUTS external application begin the Calibration.

Example:

{"Event":"DonutsCalibrationStart","Timestamp":1619784510.33227,"Host":"ORIONE","Inst":1}

y) DonutsCalibrationDone

Raised when DONUTS external application correctly End the Calibration.

Example:

{"Event":"DonutsCalibrationDone","Timestamp":1619784510.33227,"Host":"ORIONE","Inst":1}

z) DonutsCalibrationError

Raised when DONUTS external application End the Calibration with errors.

Attribute	Type	Description
DonutsError	string	Text of the error in Donuts

Example:

{"Event":"DonutsCalibrationError","Timestamp":1619784510.33227,"Host":"ORIONE","Inst":1,"DonutsError":"This is the Donuts error"}

aa) DonutsRecenterRequired

Raised when Voyager Ask to DONUTS external application to start the Recenter Task.

Attribute	Type	Description
FITPathAndName	string	Path and name of the FIT File to analyze like reference for centering

Example:

{"Event":"DonutsRecenterRequired","Timestamp":1619797004.11734,"Host":"ORIONE","Inst":1,"FITPathAndName":"C:\\prova.fit"}

bb) DonutsRecenterStart

Raised when DONUTS external application begin the Recenter.

Example:

{"Event":"DonutsRecenterStart","Timestamp":1619784510.33227,"Host":"ORIONE","Inst":1}

cc) DonutsRecenterDone

Raised when DONUTS external application correctly End the Recenter.

Example:

{"Event":"DonutsRecenterDone","Timestamp":1619784510.33227,"Host":"ORIONE","Inst":1}

dd) DonutsRecenterError

Raised when DONUTS external application End the Recenter with errors.

Attribute	Type	Description
`DonutsError`	string	Text of the error in Donuts

Example:

{"Event":"DonutsRecenterError","Timestamp":1619784510.33227,"Host":"ORIONE","Inst":1,"DonutsError":"This is the Donuts error"}

11. Workflow

- Open connection to the server
- Read Socket in a Loop and Start it in a separate Thread if possible
- You'll receive (one time) at beginning the `Version` Event FROM server
- You'll receive each 5s the `Polling` Event FROM server also if the server do not have data to send
- Read and process the events received
- Send command if needed and wait response to command, reset your polling timer when send data
- 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 (recommended) 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}