ASCOM REMOTE

Installation and Configuration

Abstract

Instructions for installing and configuring ASCOM Remote

Peter Simpson
peter@peterandjill.co.uk

Contents

1. ASCOM Remote Installation and Setup

1.1 Pre-Requisites

Please note that ASCOM Remote requires .NET Framework 4.8, which means that the operating system must be Windows 7 SP1 or later because .NET 4.8 is not available on earlier operating systems.

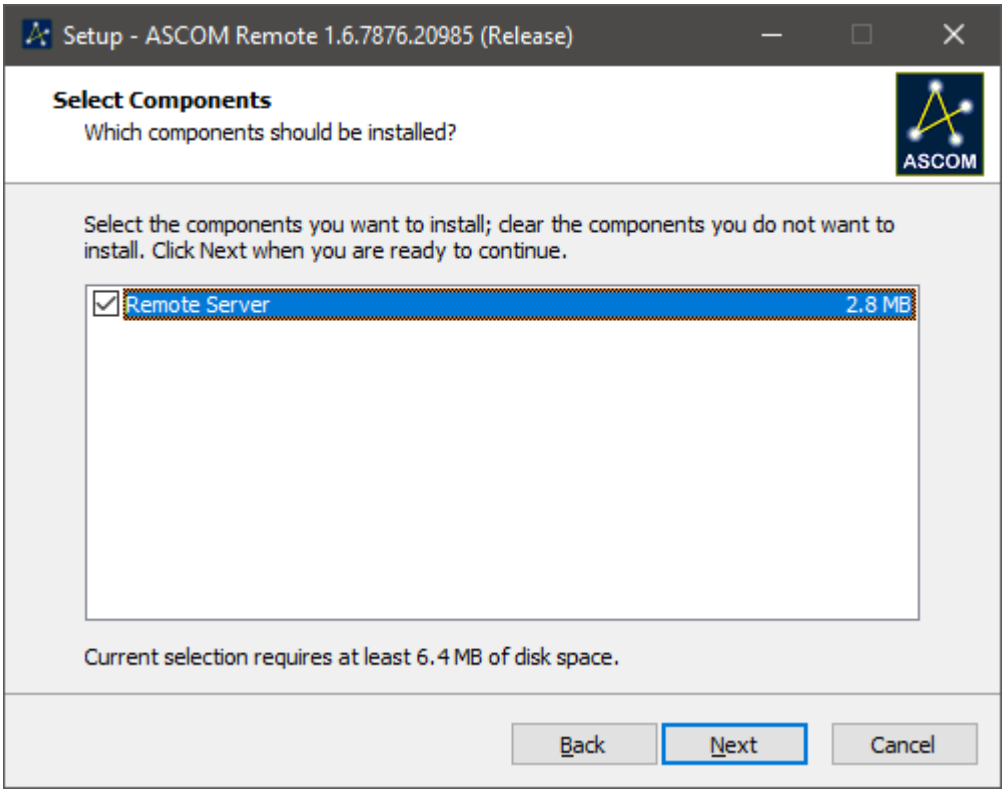
1.2 Remote Clients have been deprecated

Since Platform 6.5 provides the improved Alpaca "Dynamic Clients", the "Remote Clients" previously distributed with the Remote Server are no longer supported and have been removed from the installer.

Dynamic Clients are more advanced than Remote Clients e.g. they support Alpaca Discovery and are the strategic approach to present Alpaca devices to client applications that use COM drivers.

1.3 Installation

Please select the "Remote Server" component to install the Remote Server.



2. Configuring Dynamic Clients

The Remote Clients appear in Chooser as normal ASCOM drivers:

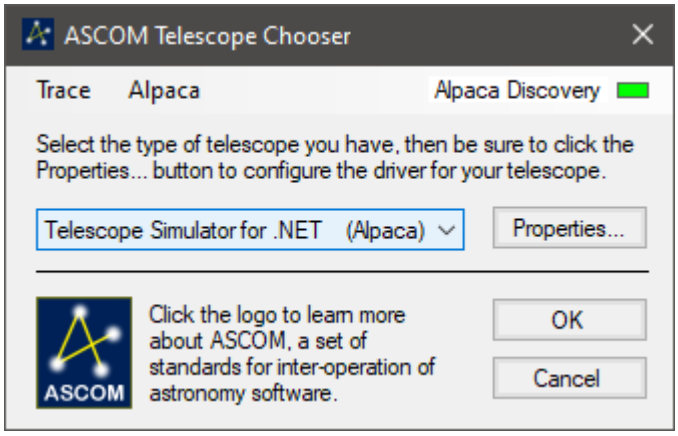


Figure 1 - Telescope Chooser showing a remote client

and can be configured through Chooser's Properties button in the usual way. Instructions on using the new Chooser functionality are provided online here.

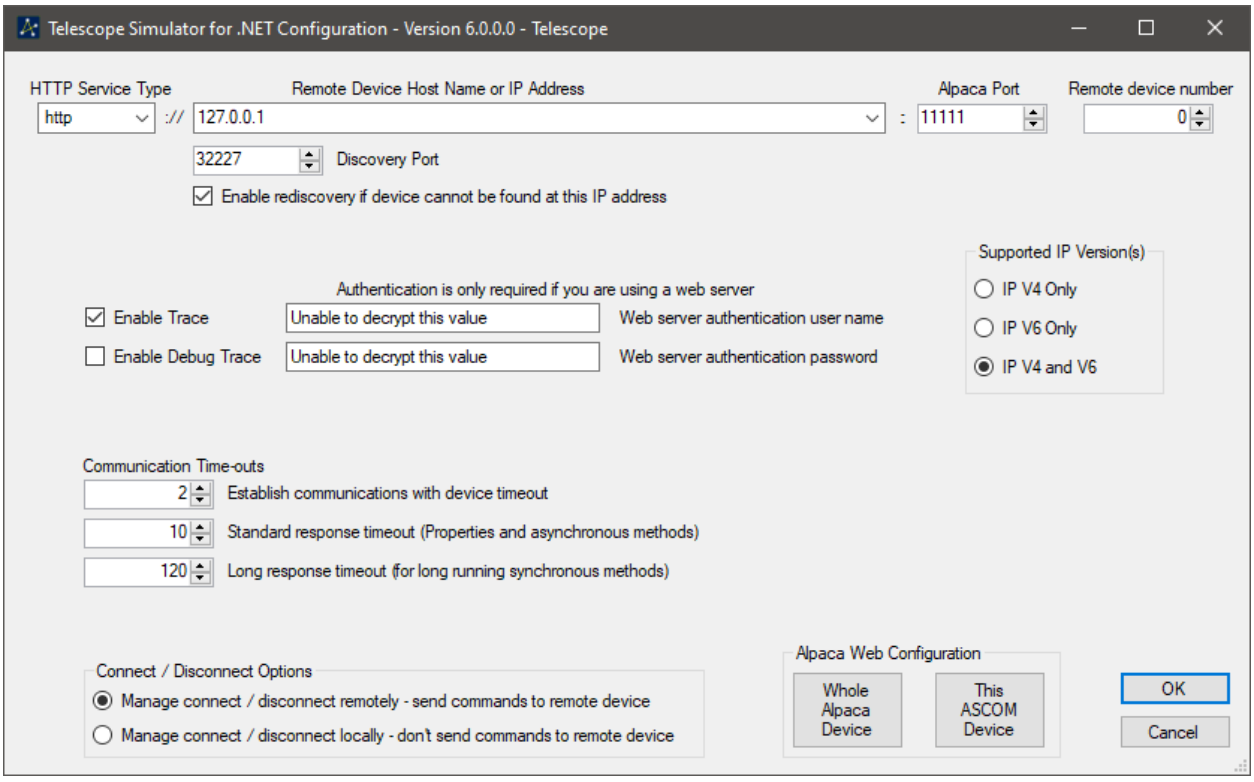


Figure 2 - Dynamic client configuration form

2.1 Common Configuration

2.1.1 HTTP Service Type, IP Address, IP Port

The service type (HTTP/HTTPS), IP address or host name and port number must match the values used when configuring the remote Alpaca device or Remote Server as appropriate.

Alpaca supports IPv4 as well as IPv6 and the address scope can be selected through the "Supported IP Versions" radio buttons.

2.1.2 Remote Device Number

The Remote Device Number needs must match the zero-based device number that is set on the Alpaca device or assigned by the Remote Server as appropriate.

2.1.3 Discovery

The Dynamic Clients can use the Alpaca discovery protocol to locate Alpaca devices. The registered discovery port is 32,227 and it is not recommended that this is changed. If you do decide to use a different discovery port, you must configure the new value on all Alpaca devices.

2.1.4 Authentication

Dynamic Clients support HTTP "Basic Authentication". If your Alpaca device requires authentication, set the username and password in the fields on the setup dialogue. Any values entered are encrypted before being persisted in the Profile.

2.1.5 Communication Timeouts

There are three communication timeouts, one for establishing an initial connection with the Remote Server the second for relatively quick response commands such as CanXXX properties and a third for slow response commands such as Telescope.SlewToCoordinates. The standard response timeout default should suit most requirements, but you may need to increase the slow response timeout depending on the longest command completion time expected under normal circumstances for your remote device.

2.1.6 Connect - Disconnect Options

Dynamic clients can be configured to send "Connected" property changes to the remove device or to respond locally without communicating with the remote device. It is recommended that connection is managed remotely. If local management is enabled, the remote device will never close.

2.1.7 Alpaca Web Configuration

Alpaca devices can optionally support HTTP web pages providing device description and device management. The "Whole Alpaca Device" button displays the main page for the whole Alpaca device (regardless of how many ASCOM devices it is presenting), while the "This ASCOM Device" button displays a configuration page dedicated to the ASCOM device is being managed in this Setup dialogue.

2.2 Camera Device Configuration

Camera devices support two additional configuration options that determine aspects of image array transfer.

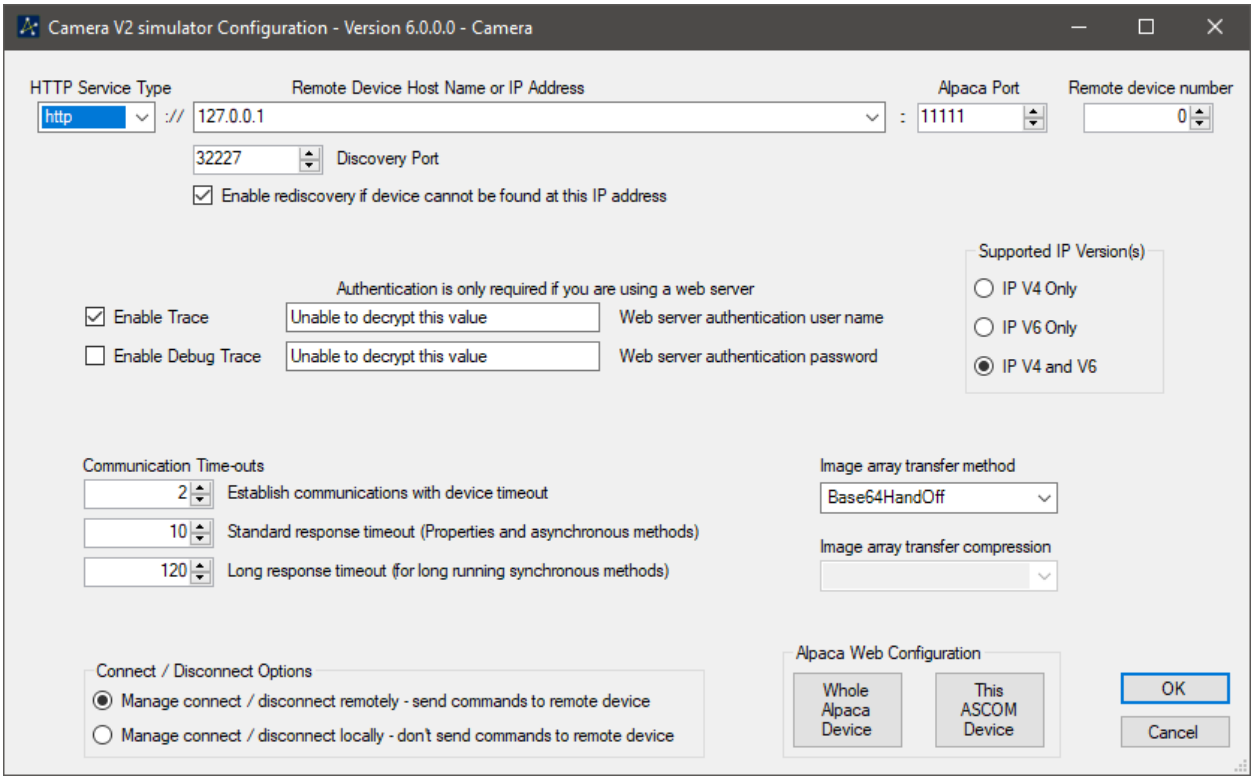


Figure 3 - Remote Server configuration dialogue

2.2.1 Image Array Transfer Method

- **JSON** - Uses the original JSON encoding mechanic per the Alpaca specification, which can be slow for large images.
- **Base64HandOff** - Requests use of a base64 handoff mechanic, which returns a small JSON response (see section 3.3) and permits downloading of a base64 encoded version of the image. In testing transfer times for 4000 x 3000 images reduced from typically 12 seconds to less than 2 seconds. Transfer compression cannot be requested in this mode because it always degraded overall timings.

2.2.2 Image Array Transfer Compression

- None - No compression will be requested
- Deflate - Deflate compression will be requested
- GZip - GZip compression will be requested
- GZipOrDeflate - Both GZip and Deflate compression will be requested, the remote device will choose which is used.

3. ASCOM Remote Server

The Remote Server application start shortcut is located in the Start Menu / ASCOM Remote folder.

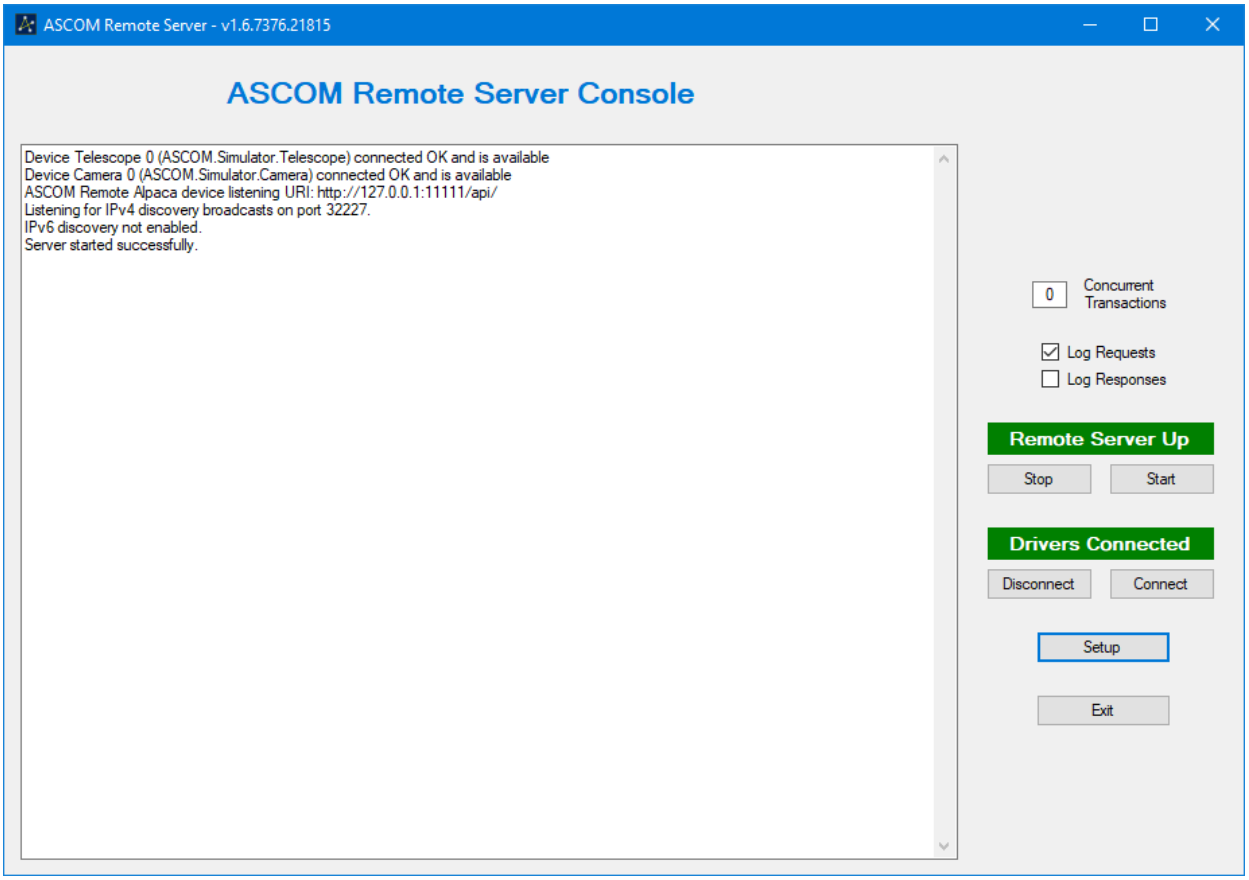


Figure 4 - Remote server console

When the Remote Server starts it will list:

- The configured devices
- The URI on which it is listening for requests
- The Alpaca discovery port on which it is listening

Remote server configuration is effected through the Setup button.

For testing its fine to run the driver and the remote server on the same PC and to use 127.0.0.1 as the IP address for both clients and server

3.1 Setup - Device Configuration

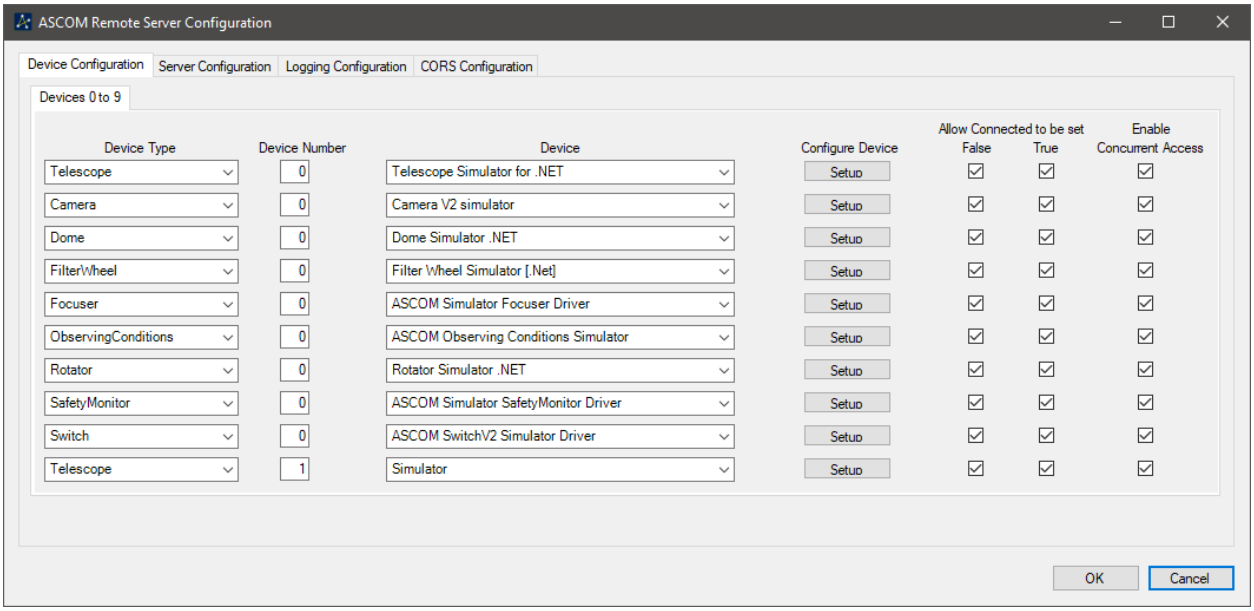


Figure 5 - Device configuration dialogue

By default, the remote server initially supports up to 10 devices. If more than 10 devices are required, additional tabs supporting a further 10 devices can be added by setting the "Maximum number of served devices" control on the Server Configuration tab.

3.1.1 Device Selection

To set up a device to be remotely served, first select the type of device in one of the "Device Type" drop-down boxes, then select its driver from the corresponding "Device" drop-down box. Make sure that all unused "Device Type" dropdowns are set to "None".

"Device Numbers" are automatically assigned as device types are selected and relate to the number of devices of that specific device type that are configured. E.g. the first focuser driver that is configured will be focuser device "0" while the second focuser device will be focuser device "1" etc.

The configured "Device Number" and "Device Type" uniquely identifies a remote device and **it is these that must be configured in the remote client** to specify the required remote device.

3.1.2 Device Configuration

The device's configuration screen can be accessed through its "Setup" button.

3.1.3 Connected State Management

The "Allow Connected" check boxes determine whether "Set Connected True" and "Set Connected False" requests will be sent to the device, which enables a device to be maintained in a connected state even if a client disconnects. When the "Connected" check boxes are unset, client drivers will see Telescope.Connected changing state as they expect, but the state of the remote device will not change.

This feature will be of value in multi-client environments where the observatory operator can prevent devices being commanded offline by one client while still in use by another.

3.1.4 Enable Concurrent Access

These check boxes control whether the Remote Server will send commands to each device as they are received, even if previous commands have not completed, or whether the Remote Server will queue the commands and send them to the device one at a time, in the order in which they were received.

3.2 Setup - Server Configuration

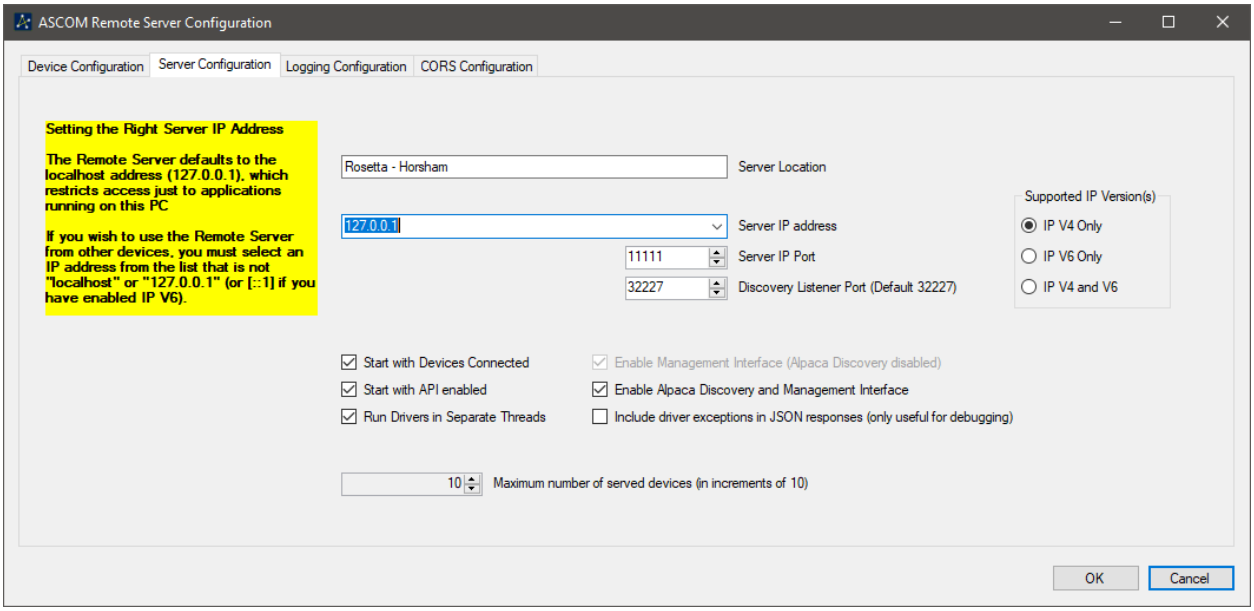


Figure 6 - Remote server configuration dialogue

3.2.1 Supported IP Version(s)

The Remote Server can bind to addresses from both the IPv4 and IPv6 families. Families are enabled by selecting the appropriate radio button in the Supported IP Versions group to the right of the dialogue.

"IP V4 Only" is the recommended setting because this is expected to be the dominant family for some time to come and, at the time of writing in March 2020, there are no known Alpaca clients that only communicate over IPv6.

3.2.2 IP Address and Port

The Server IP Address dropdown and Server IP Port selector enable you to select the IP address and port number on which the server will listen. The IP address list will be pre-populated with all the available network addresses, of the selected IP version(s) on the host PC, plus an IPv4 specific "localhost" entry when appropriate.

If you have more than one network interface and select the "All IP Addresses" option in the Server IP Address dropdown, the Remote Server will listen on all listed network addresses.

3.2.3 Discovery

The Remote Server supports the Alpaca Discovery Protocol and will respond to discovery broadcasts on every IPv4 address that is configured in the "Server IP address" field. In addition, it will respond to IPv6 discovery multicasts on every IPv6 link local address that is configured plus the "::1" IPv6

localhost address if appropriate. By design, to protect the Remote Server from being used in denial-of-service attacks, there will be no response to discovery packets on IPv6 global unicast addresses.

3.2.4 Server Location and Management Interface

The management interface conforms to the Alpaca Management API standard as documented here: Alpaca Management API Specification

It returns information on the remote server as a whole, including the "Server Location" field and a list of the devices configured on the Device Configuration tab. The location field can contain descriptive text such as a physical location or a PC or VM machine name.

The management interface is enabled or disabled through the "Enable management interface" check box.

3.2.5 Remote Server Startup

Whether the Remote Server starts with devices unloaded or loaded and with the listening URI enabled or disabled can be controlled from the "Auto connect devices" and "Start with API enabled" checkboxes.

3.2.6 Drivers in Separate Threads

The "Run Drivers in Separate Threads" checkbox chooses between:

- **Enabled**: Runs each driver in its own thread with an exclusive Windows event loop. (Default)
- **Disabled**: Runs all drivers on the Remote Server's UI thread sharing a common Windows event loop.

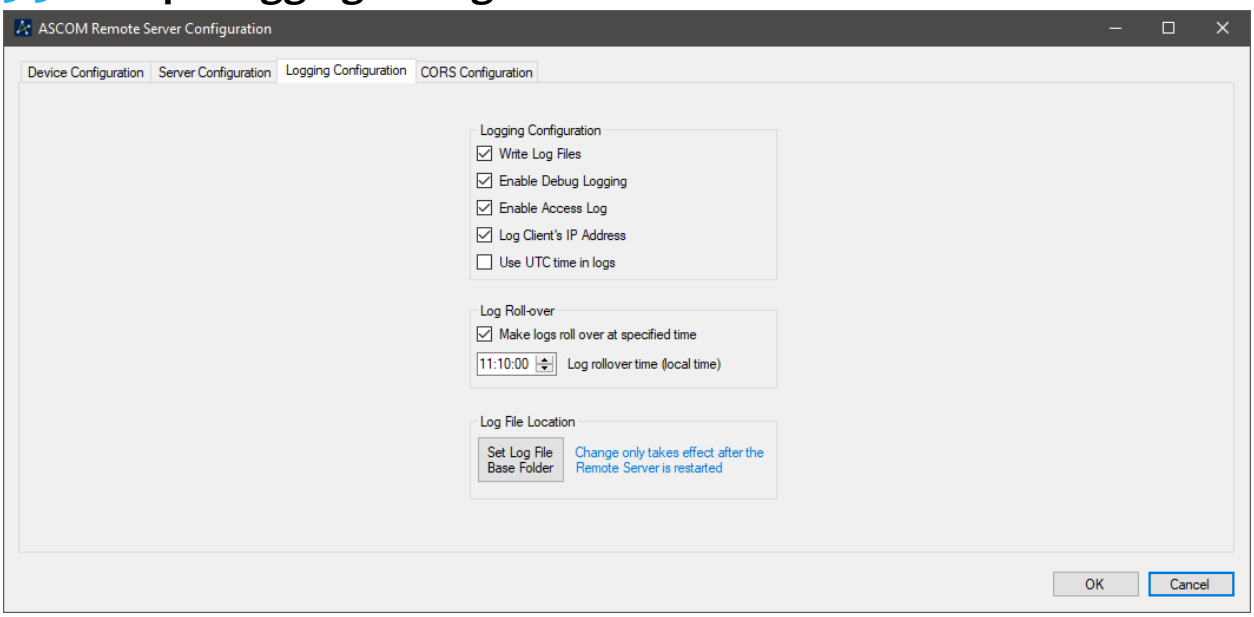
Running drivers in their own threads is the preferred mode of operation because it provides greater isolation of driver issues from other drivers and from the Remote Server itself. There are currently no known downsides to this approach; the "run all on the main thread" option, however, is provided as a fall back in case of issues arising when using separate threads.

3.2.7 Maximum Number of Served Devices

By default, the Remote Server is configured to present up to 10 devices. However, it can serve up to a maximum of 100 devices by setting the "Maximum Number of Served Devices" to the required number.

Additional tabs of 10 devices will appear on the "Device Configuration" page as the maximum number of served devices is increased.

3.3 Setup - Logging Configuration



3.3.1 Logging and Debug Logging

These options control whether log files are created and the amount of detail they contain.

3.3.2 Access Log

The Access Log records one line for each request, for audit purposes, which includes the client IP address and request. The log does not include any indication of the outcome of the request.

3.3.3 Use UTC Time

This option specifies whether UTC or Local time is recorded in the logs.

3.3.4 Log Roll-over

By default, logs will continue for the lifetime of the Remote Server instance. To support long running Remote Server instances, this option instructs the Remote Server to close the current log and start a new one when the next log entry is made following the set time.

3.3.5 Log File Location

This option allows the Remote Server's log file location to be changed from its default Documents\ASCOM value.

3.4 Setup - CORS Configuration

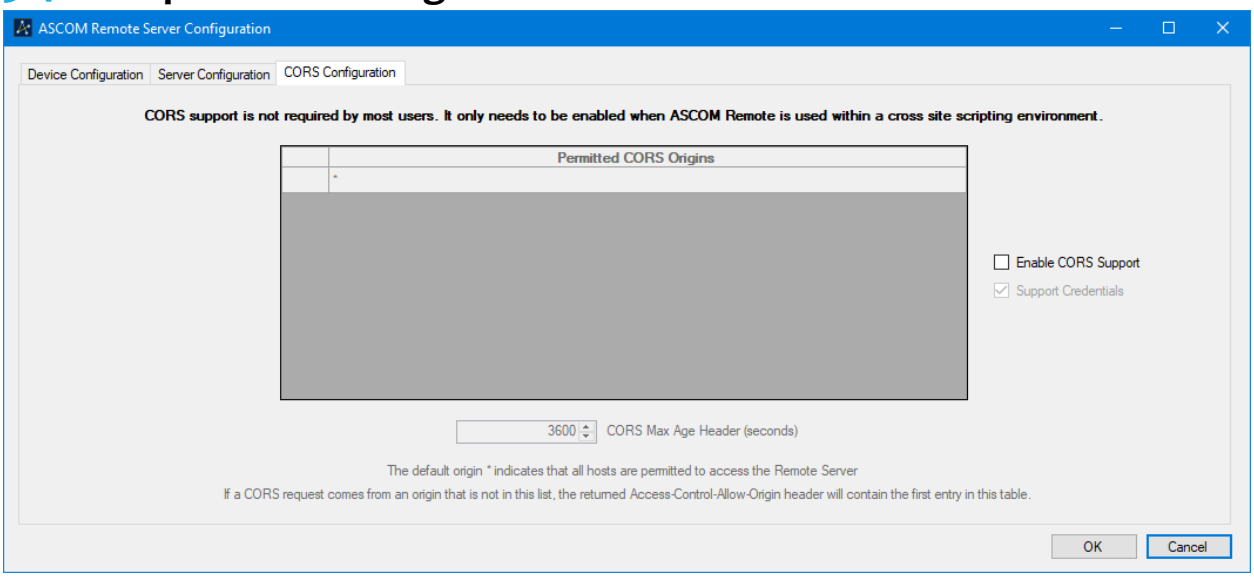


Figure 7 - CORS configuration dialogue

Most people will not need CORS support, it is only required when the ASCOM Remote Server is incorporated as part of a web application that is accessed through a browser.

The default CORS configuration, when enabled, uses a permitted origin of "*", which permits access from all hosts.

The CORS implementation flow diagram is shown over page in Figure 8 - CORS flow diagram.

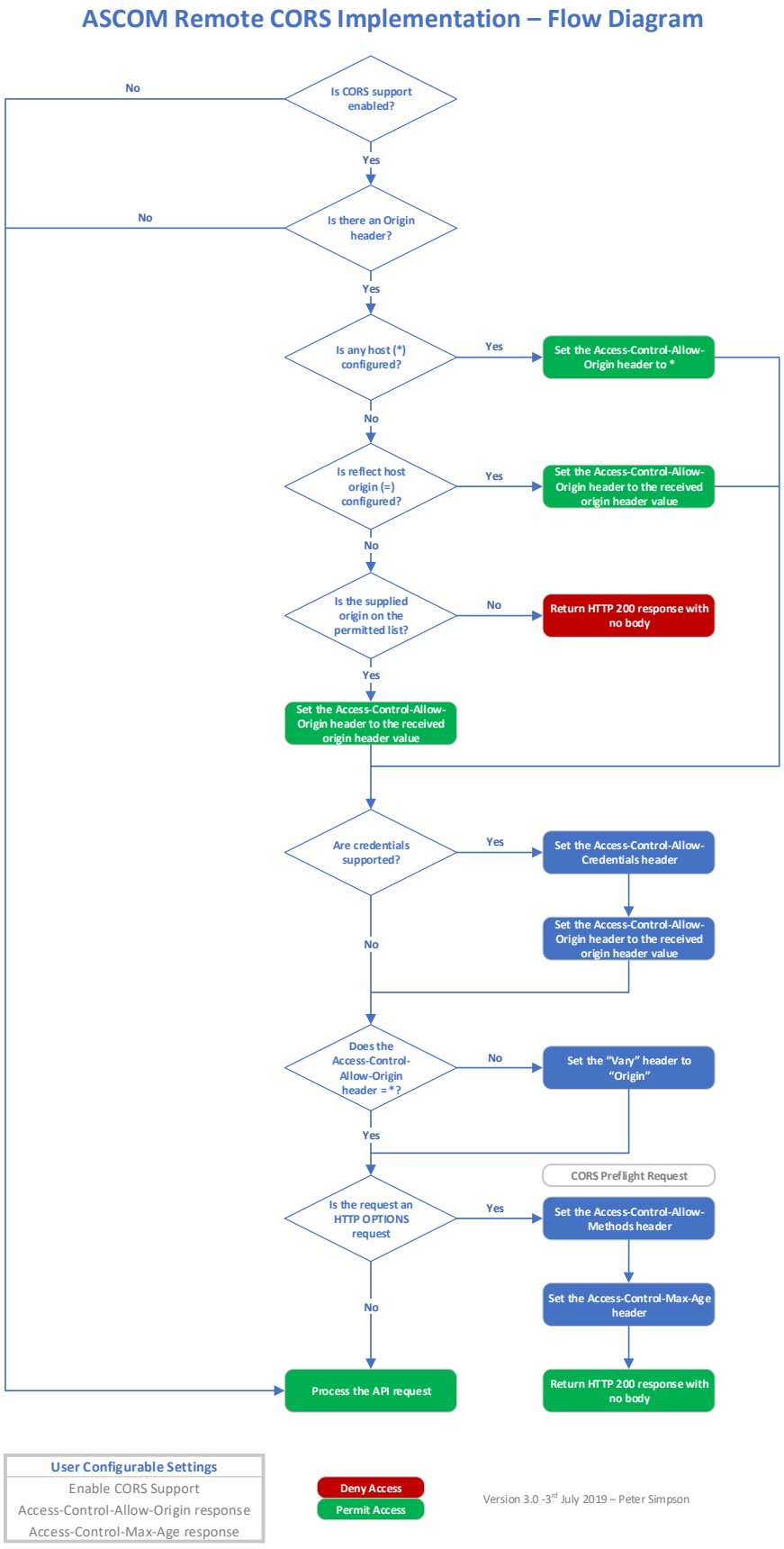


Figure 8 - CORS flow diagram

3.5 Using Camera.ImageArray Base64 Handoff Mode

The ASCOM Remote camera client has built-in support for the base64 handoff mode, which can be enabled through the client setup dialogue.

This section will be of interest to developers creating their own applications e.g. in Python who want to take advantage of the speed increase that the base64 handoff mode provides.

To request use of the base64 handoff mode, if available, the client should add this HTTP header to the HTTP GET /api/v1/camera/x/imagearray request:

`base64handoff = true`

The Remote Server will then request the image from the device and, after it is available, will return a small JSON response similar to Figure 9 - Small base64 handoff JSON response. The JSON response will have the same HTTP header:

`base64handoff = true`

indicating that it supports the base64 handoff mechanic. If the header is absent, the JSON response should be interpreted as the large JSON response containing the image array data that is described in the Alpaca API Specification.

```
{
"Type": 2,
"Rank": 2,
"Dimension0Length": 4,
"Dimension1Length": 4,
"Dimension2Length": 0,
"ClientTransactionID": 0,
"ServerTransactionID": 182,
"ErrorNumber": 0,
"ErrorMessage": ""
}
```

Figure 9 - Small base64 handoff JSON response

The Type, Rank, ID and Error fields are as specified for the Alpaca ImageArray response

The base64string is obtained by an HTTP GET the endpoint:

`/api/v1/camera/x/imagearraybase64`

where x is the camera device number as used in the original GET to the device's imagearray endpoint.

Once received, it is the client's responsibility to decode the base64 string into an array whose dimensions are given in the small JSON response DimensionXLength fields.