

Network Agent - Server

Part of Cindev's Agent Bundle



Compatible Control4 Systems:

Designed to work with OS 3.x+

Driver Download, Change Log & Documentation:

<https://drivercentral.io/platforms/control4-drivers/utility/network-agent/>

Content

- [Overview/Features](#)
- [Dealer Notes](#)
- [Agent Setup](#)
- [Agent Properties](#)
- [Server Settings](#)
- [Events](#)
- [Conditionals](#)

- [Commands](#)
 - [Support](#)
 - [License Purchase & Assignment](#)
 - [Showroom and Free Trial](#)
 - [Warranty & Disclaimer](#)
 - [Developer Information](#)
-

Overview/Features

You can now utilize networking protocols to communicate directly with APIs and other endpoints on the network directly from your Control4 system. The Network Agent Server driver provides you with tool to start a HTTP, TCP or UDP server which listens for requests and replies to the requests accordingly.

- Send custom HTTP, TCP and UDP replies
- Global headers can be added to every HTTP reply
- Set Default Responses for every type of server
- Various ways to secure the server using the Access Control setting
- Full programming control via conditionals for every portion of a message's request and response

To save money and add other cool features to your projects, consider purchasing as part of the Agent Bundle.

Dealer Notes

- All headers must use the following format:
header1=value1&header2=value2...
 - Driver contains a buffer between each request to property handle Request and Response events
 - Variables are cleared after each event is processed. To prevent clearing variables turn on Debug Mode in the driver. This is useful in order to determine why a command or conditional is failing.
-

Agent Setup

1. Select Server Type
 2. Customize Advanced Settings as needed (see Server Settings below)
 3. Use the programming section of composer to trigger actions based on server's requests and responses
-

Agent Properties

- *Cloud Status*: Displays driver license state
 - *Driver Status*: Displays driver related information
 - *Driver Version*: Displays driver version
 - *Driver Actions*:
 - *View History*: View server requests/response history on the lua tab
 - *Automatic Updates*: Allows the *Cloud Driver* to automatically update the driver when a new version is available
 - *Debug Mode*: Displays additional information on the lua tab for debugging purposes
-

Server Settings

- *Type*: select the type of server to start. Selecting '(Select)' will turn off the server.
- *Host/IP*: displays the address of created server
- *Advanced Settings*:
 - *HTTP Headers*: (HTTP Server Only) global headers that will be used in every HTTP response
 - *Default HTTP Response*: (HTTP Server Only) Default HTTP response if not otherwise set in programming
 - *Default TCP/UDP Response*: (TCP/UDP Server Only) Default TCP/UDP response if not otherwise set in programming
 - *Access Control*: various ways of securing server
 - *Whitelist*: list of allowed IP addresses
 - *Blacklist*: list of blocked IP addresses

- *Basic Auth*: utilizes HTTP's basic auth to authenticate using a username and password
 - *Token*: utilizes a bearer token for authentication. Requests require a "Bearer" header with the token value.
-

Events

- **Request Received** Triggered every time the driver receives a request
- **Response Sent** Triggered every time the driver sends a response after receiving request

In order to leverage full control of the events utilize conditionals as described in the following section.

Conditionals

Conditionals utilize driver variables to uniquely process each request and response. Each conditional should be used with the corresponding event as follows:

- **Request Conditionals**

- *Request Origin*: Used with HTTP TCP and UPD messages
- *Request Method*: Used with HTTP messages
- *Request Path Command* Used with HTTP TCP and UPD messages
- *Request Headers* Used with HTTP messages
- *Request Query* Used with HTTP messages
- *Request Body* Used HTTP messages

- **Response Conditionals**

- *Response Status* Used with HTTP messages
 - *Response Headers* Used with HTTP messages
 - *Response Body* Used with HTTP messages
 - *Response Command* Used with TCP and UPD messages
-

Commands

- **HTTP Response** Send a HTTP response with these custom parameters:
 - *Status*: used to set status of the response
 - *Headers*: provide additional headers with response
 - *Body*: provide body with response
 - **TCP/UDP Response** Send back a TCP or UDP command:
 - *Command*: respond to the TCP request with a command
-

Support

If you require additional driver assistance or are having some issues please visit our help desk: <https://help.drivercentral.io>

Showroom and Free Trial

All Cindev drivers for Control4 come with a 7-day trial. Simply add the DriverCentral licensing driver, *Cloud Driver*, along with this driver and you can fully test the driver before purchasing. Including Cindev drivers in your drivercentral showroom project will give you unlimited use without purchase.

Warranty & Disclaimer

www.cindev.com/terms-and-conditions

Developer Information

brought to you by:
Cinegration Development, LLC



www.cindev.com
www.drivercentral.io/cindev/

We are always looking to improve our drivers.
Please send your suggestions to: info@cindev.com