

HTTPIN Universal Connector

12/18/2023 11:42 am EST

Contents

[Purpose](#)
[Process Overview](#)
[String Format](#)

Purpose

HTTPIN TCP connectors accept GET/POST/PUT requests.

HTTPIN connectors require elevated permissions. MediaGateway uses netsh.exe to bind the certificate to the port. The UC will work in console mode or standalone mode if run as administrator.

If running in a standalone mode not as administrator the user will see errors such as:

Error: Error trying to bind the certificate to 0.0.0.0:9001. Has exited: True, Exit code: 1

{ConnectorName}: Startup: Not Connected

{ConnectorName} listener not initialized.

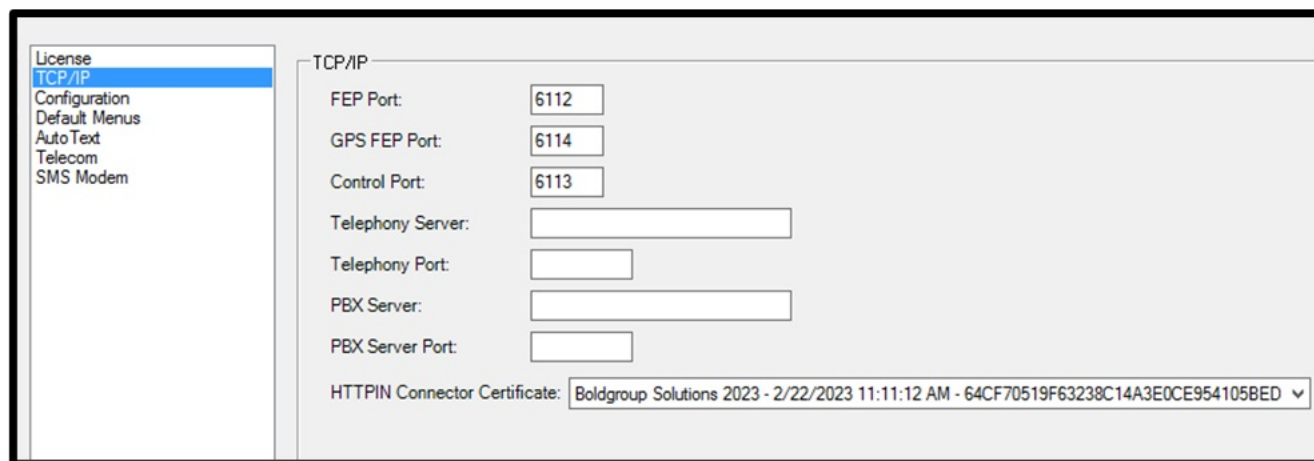
When running the MediaGateway as a service using the Local Service Account there should be no errors.

[Return to Contents](#)

Process Overview

Import a server certificate into Local Computer->Personal in Certificate Manager if one does not already exist. This certificate can be a self-signed certificate if none else is available.

Open MediaGateway in Console or as Standalone as administrator. Navigate to Options and select TCP/IP. In the drop-down for HTTPIN Connector Certificate select the certificate you choose to use. Only valid certifications will show in the drop-down.

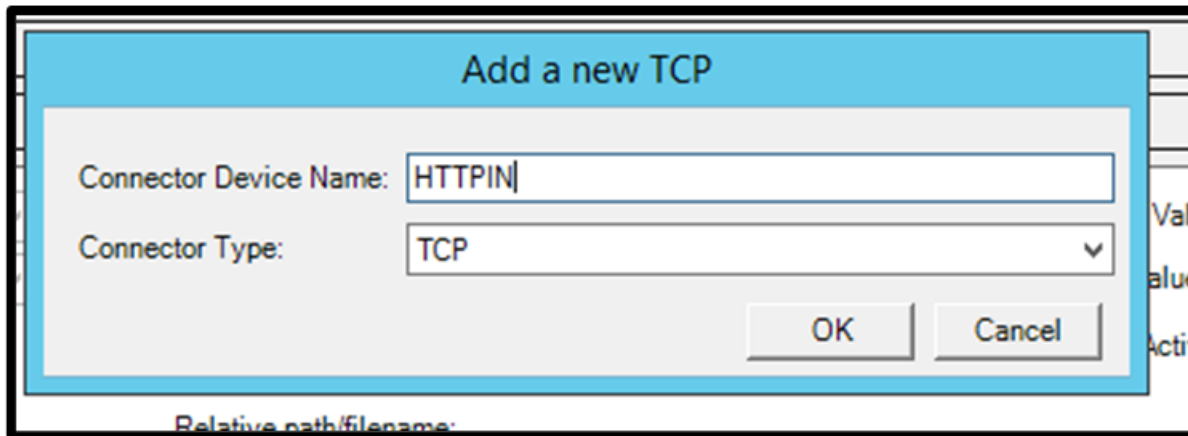


The screenshot shows the MediaGateway configuration window with the 'TCP/IP' tab selected. The left sidebar contains a menu with options: License, TCP/IP (selected), Configuration, Default Menus, Auto Text, Telecom, and SMS Modem. The main area displays the following settings:

Field	Value
FEP Port:	6112
GPS FEP Port:	6114
Control Port:	6113
Telephony Server:	
Telephony Port:	
PBX Server:	
PBX Server Port:	
HTTPIN Connector Certificate:	Boldgroup Solutions 2023 - 2/22/2023 11:11:12 AM - 64CF70519F63238C14A3E0CE954105BED

Next navigate to Connectors.

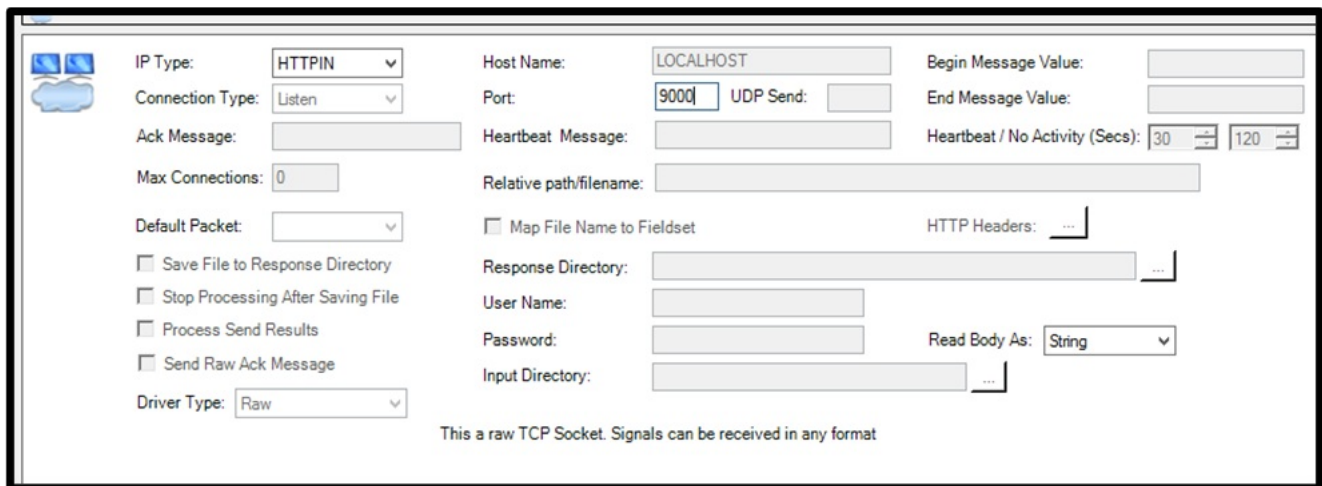
Select Add to add a new connector.



Select the Connector Type as TCP.

Input the Port that will be used. The port will need to be opened in the Firewall and rules set there as to who can connect on that port from outside sources.

Set the Read Body As either String or KeyValuePair. Save and do a Reload/Restart to verify changes are saved.



[Return to Contents](#)

String Formats

Read Value As: String

Read as String will read PUT/POST body as string and not manipulate it. This is good for XML/JSON which is structured how MediaGateway expects it.

MG expects XML/JSON to have a root node and a signals array (they don't have to be specifically called "Root" and "Signals").

Properly formatted JSON:

```
{
  "Root": {
    "Signals": [
      {
        "TxId": "4001",
        "EvCode": "BA",
        "Latitude": "N53.415969",
        "Longitude": "W1.401091",
        "Speed": 0,
        "Heading": 0
      },
      {
        "TxId": "4001",
        "EvCode": "BA",
        "Latitude": "N53.415969",
        "Longitude": "W1.401091",
        "Speed": 0,
        "Heading": 0
      }
    ]
  }
}
```

Properly formatted XML:

```
<?xml version="1.0" encoding="utf-8"?>
<Root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" PositioningMode="Gps" MovementStatus="Moving"
PowerOffMethod="PowerOffByManual" PowerStatus="PowerOn" Imei="867967061221460" SimCardNumber="0"
WorkingMode="1" UserPassword="0000" SoftwareVersion="0.9" BatteryLevel="51" PacketType="AUT"
DateAndTime="071217224908" GsmBlindAreaReportPacket="false">
  <Signals EventCode="*SPLOCAT">
    <Latitude>N53.415969</Latitude>
    <Longitude>W1.401091</Longitude>
    <Speed>0.00</Speed>
    <Heading>0.00</Heading>
  </Signals>
  <Signals EventCode="*SPOOFNC">
    <Latitude>N53.415969</Latitude>
    <Longitude>W1.401091</Longitude>
    <Speed>0.00</Speed>
    <Heading>0.00</Heading>
  </Signals>
</Root>
```

Read Value As: KeyValuePair

Read as KeyValuePair will read PUT/POST body as string and try to convert it to key value pairs.

You could pass in the body of PUT/POST this:

```
txid=4001&evcode=BA&zone=7
```

and we would parse it into this so MediaGateway can process it:

```
txid=4001{cr/lf}
evcode=BA{cr/lf}
zone=7{cr/lf}
```

You could pass a JSON object (no root node, no signals array) like this:

```
{
  "txid":"4001",
  "evcode":"BA",
  "zone":"3"
}
```

and we would parse it into this so MediaGateway can process it:

```
txid=4001{cr/lf}
evcode=BA{cr/lf}
zone=7{cr/lf}
```

NOTE: GET requests convert ONLY QUERY PARAMETERS to key value pairs.

For example, if we get this: <https://bglptp30.boldgroup.int/1/2/3?txid=4001&evcode=BA&zone=7>

MG will only receive:

```
txid=4001{cr/lf}
evcode=BA{cr/lf}
zone=7{cr/lf}
```

[Return to Contents](#)