SDP Specification

Joah George — August 3, 2023

The Smart Device Protocol (SDP) is a text-based protocol designed to manage multiple smart devices on a server. The basic format of the protocol is:

<Operation>~~~[Device Flag]~~~[Device Names]~~~[Value]

A note on wording:

* “must” is a non-negotiable
* “should” is a recommendation
* “irrelevant” means it does not matter what the value is

# Operation:

* PULL – this will get information about the device(s) listed in the [Device Names].
  + Used by Client
  + [Device Flag] must be positive
  + [Device Names] should not be empty
  + [Value] is irrelevant
* PUSH – this will set the device(s) listed in [Device Names] to the value in [Value]
  + Used by client
  + [Device Flag] should not be 0
  + [Device Names] should not be empty
  + [Value] should not be empty
* REPLY - this signifies that the server is responding to a client’s PUSH or PULL operation. The result of the operation is stored in the [Value] field. A REPLY will be sent for each device listed in the [Device Names] field
  + Used by server
  + [Device Flag] should be a power of 2
  + [Device Names] should be a single device name
  + [Value] should not be empty
* ERROR – this signifies that the server is responding the a client’s PUSH operation, but there was some error is setting the device’s status
  + Used by server
  + [Device Flag] should be a power of 2
  + [Device Names] should be a single device name
  + [Value] should be the error message
* EXIT – this will stop the server by closing the client connection and then closing the server socket. To be used after client is done making changes
  + Used by client
  + [Device Flag] is irrelevant
  + [Device Names] is irrelevant
  + [Value] is irrelevant

# Device Flag:

The [Device Flag] holds information about which devices will be accessed or have been accessed. Currently, the flag only needs 2 bits – one bit for each device type on the network. However, the flag is expressed in decimal form rather than binary to stay consistent with the text-based nature of the protocol.

Each bit represents a single device type, so when all devices are needing to be accessed, all bits will be 1. If only a single device type needs to be accessed, then only the device type’s corresponding bit will be set to 1 and the others will be set to 0.

## Ex:

Device Types: Thermostat and LightSwitch

Device Flag: 2 bits

|  |  |
| --- | --- |
| Thermostat (Upper) Bit | LightSwitch (Lower) Bit |
| 0 | 0 |

If both device types are needing to be accessed in a PUSH message, then the device flag would be 11 in binary and 3 in decimal.

If only LightSwitches have been accessed in a REPLY message, then the device flag would be 01 in binary and 1 in decimal.

## Use as a client and server’s understanding (in PUSH and PULL operations):

When a client sends the [Device Flag], it intends to tell the server which device types the client is trying to access. This helps in reducing the time to check devices by only checking certain device types when only certain device types are being accessed.

## Use as a server and client’s understanding (in REPLY and ERROR operations):

When a server sends the [Device Flag], it intends the tell the client what device type the [Device Names] field contains. The [Device Flag] should only contain a single device type.

# Device Names:

The [Device Names] field holds the names of the device(s) that will be or have been accessed. The [Device Names] field is the list of device names using “^^^” (without the quotes) as the delimiter and has the format:

[Device Name]^^^[Device Name]^^^…^^^[Device Name]

The [Device Names] field can be as long as needed and the [Device Name]s can be as long as needed.

## Use as a client and server’s understanding (in PUSH and PULL operations):

When the client sends the [Device Names] field, it intends to tell the server which devices need to be affected by the operation. All device types specified in the [Device Flag] will be checked for the name(s) in [Device Names].

## Use as a server and client’s understanding (in REPLY and ERROR operation):

When the server sends the [Device Names] field, it intends to tell the client which device the [Value] field pertains to.

# Value:

The [Value] field indicates what the status of the device should be, is, or what it could not be. The [Value] field is a string that holds information.

## Use as a client and server’s understanding (in PUSH operation):

When the client sends the [Value] field, it intends to tell the server what the status of the devices listed in the [Device Names] field should be set to.

## Use as a server and client’s understanding (in REPLY and ERROR operations):

When the server sends the [Value] field, it intends to tell the client the status of the device (REPLY) or that the device could not be set to a value (ERROR).