D-BOX TECHNOLOGIES INC.

Monitoring Service Communication

TCP/IP Communication API

129-975-0001-EN7

Monitoring Service Communication

Connection

Monitor Service is a Windows service running on a computer. A connection can be established by opening a TCP/IP connection on the computer on 127.0.0.1 (local) or its LAN IP address on port 40001 (default port).

Communication format (API)

Each request will generate a reply. Replies contain the original request information so that they can be matched to the request. Requests and replies must end by [CR/LF] (0x0D followed by 0x0A). No whitespaces or new lines can be used in requests and replies, except for [CR/LF] that ends each message.

Error codes are used when the request is invalid.

<Reply Cmd="CmdName" Error="x" Message="abc" />[CR/LF]

Unknown command: -1
Invalid parameter (including missing mandatory parameter): -2
No request (No *Request* node): -3
No command (no *Cmd* attribute): -4
Invalid request: -5
Communication Unit not available: -6

Communications are always XML based. The order of the nodes and attributes should not be used when processing the data, only the XML structure itself should be used (ie: the attribute and node names must be used, not the indexes).

1. New device notification

Usually a reply is only sent following a request except for device notifications (so that polling is not required). When a new communication unit is connected or disconnected, a message (without a request) is sent as a notification. Based on the requirements, a GetLayout request could be sent by the client after receiving a notification.

Answer

<Notification Cmd="LayoutUpdate" Type="#" CommUnit="xyz" />[CR/LF]

Description

- Notification.Type: [int] Change type (1 = comm unit insertion, 2 = comm unit removal, 3 = hardware connected or disconnected from comm unit)
- CommUnit.Id: [string] Unique ID (includes serial number for KCU, audio name for KAI…) of the device that was inserted or removed.

2. Get protocol version

This command returns the protocol version (not the service version). Future protocol versions will be backward compatible with previous versions. However, it can be useful to retrieve the protocol version to know if a new feature is supported (new command, new parameter…).

Request

```
<Request Cmd="GetVersion" />[CR/LF]
```

Description

- Cmd: [string] Command name

Answer

```
<Reply Cmd="GetVersion" Error="0">
        <Version Major="#" Minor="#" />
</Reply>[CR/LF]
```

Description

- Version.Major: [int] Major version
- Version.Minor: [int] Minor version

3. Get equipment layout

This command returns the layout of the hardware connected to the communication units.

Request

```
<Request Cmd="GetLayout" CommUnitId="xyz" />[CR/LF]
```

Description

- Cmd: [string] Command name
- CommUnitId: [string] (optional) Filter for a specific communication unit

Answer

```
<Reply Cmd="GetLayout" CommUnitId="xyz" Error="0">
        < CommUnit Id="xyz" TypeId="#" TypeName="abc" Version="abc" >
                <Platform Index="#" />
                        <Acm Index="#" Id="abc" TypeId="#" TypeName="abc" Version="abc">
                                <Actuator Index="#" Stroke="xyz" RawStroke="xyz" Id="abc"
                                Model="VID.PID" TypeId="xyz" TypeName="abc" Location="(x,y)"
                                AcmMotorSlot="#" Version="abc"  />
                                 …
                                <Actuator … />
                        </Acm>
                        …
                        <Acm >…</Acm>
                </Platform>
                …
                <Platform>… </Platform>
        </ CommUnit >
        …
</Reply>[CR/LF]
```

Description

- CommUnit.Id: [string] Unique ID (includes serial number for KCU, audio name for KAI…) of the device.
- CommUnit.Name [string] Device name
- CommUnit.TypeId: [int] Device type unique ID (KAI-1P = 1, KCU-1P = 2…)
- CommUnit.TypeName: [string] Device type name (KAI-1P, KCU-1P)
- CommUnit.Version: [string] (optional) Device version details (if supported by hardware)
- CommUnit.Serial: [string] Communication device serial number
- CommUnit.ParameterSupport [int] Indicates the support level for hardware parameters. Supported = 1, Unsupported = 0. (Currently, only KCU supports hardware parameters)
- Platform.Index: [int] Platform number
- Acm.Index: [int] ACM index
- Acm.Id: [string] (optional) ACM serial number (if supported by hardware)
- Acm.Serial: [string] (optional) ACM serial number (if supported by hardware)

- Acm.TypeId: [int] ACM Type unique ID (ACM = 1, ACM-II (Master) = 2, ACM-II (Slave) = 3)
- Acm.TypeName: [string] Description of the ACM type
- Acm.Version: [string] Version number
- Acm.Model: [int] ACM Model ID
- Acm.ModelName: [string] ACM Model friendly name
- Actuator.Index: [int] Actuator number (0-…)
- Actuator.Stroke: [float] Actuator stroke in millimeters
- Actuator.RawStroke: [int] Integer actuator stroke (internal use)
- Actuator.Id: [string] (optional) Actuator unique ID (if supported by hardware)
- Actuator.Model: [string] Actuator model (vendor ID and product ID)
- Actuator.TypeId: [int] Type of actuator(Vertical = 1)
- Actuator.TypeName: [string] Description of the actuator type
- Actuator.Location: [string] (optional) Actuator emplacement on the platform (if it can be retrieved). For a vertical actuator, the format is "(x,y)" where x is a float from -1 (left) to 1 (right) and y is a float from -1 (back) to 1 (front). Values could be (0,1) for front center or (1,-1) for rear right for example.
- Actuator.Axis: [int] Mask value that determines the axis assigned to this actuator. The values are: Roll = 1, Pitch = 2, Heave = 4, Yaw = 8, Sway = 16, Surge = 32. An actuator can support more than just one axis.
- Actuator.AxisName: [string] Axis friendly name assigned to this actuator. Can be a combination of axes depending on the mask value defined previously.
- Actuator.AcmMotorSlot: [int] (optional) Motor slot of the actuator on the ACM (if supported by hardware). Starts at 0.
- Actuator.EncoderType: [int] Mask value that determines, for its position, whether this Actuator uses an absolute or relative encoder (Status<<Position). Ex.: For actuator in slot 1 (the second slot), if the encoder is absolute, the value of 2 (1<<1) is present in EncoderType.
- Actuator.EncoderTypeName [string] Encoder type friendly name
- Actuator.ConfiguredModel [float] Value, formatted VendorId.ProductId, for this actuator currently configured in the corresponding ACM
- Actuator.ConfiguredModelName: [string] Friendly name of this actuator currently configured in the corresponding ACM
- Actuator. Model [float] (optional) Value, formatted VendorId.ProductId, for this actuator currently connected in the corresponding ACM
- Actuator. ModelName: [string] (optional) Friendly name of this actuator currently connected in the corresponding ACM
- Actuator.Version: [string] (optional) Version of the actuator (if supported by hardware)

4. Get equipment status

This command returns the status of the hardware connected to the communication units.

Request

```
<Request Cmd="GetStatus" CommUnitId="xyz" />[CR/LF]
```

Description

- Cmd: [string] Command name
- CommUnitId: [string] (optional) Filter for a specific communication unit

Answer

```
<Reply Cmd="GetStatus" CommUnitId="xyz" Error="0" >
        <CommUnit Id="xyz">
                <Field Id="#" Name="abc" Value="xyz" Type="abc" Category="#" Description="abc"
                Unit="abc"  />
                …
                <Platform Index="#" />
                        <Field Id="#" Name="abc" Value="xyz" Type="abc" Category="#"
                        Description="abc" Unit="abc" />
                        …
                        <Acm Index="#" Id="abc" />
                                <Field Id="#" Name="abc" Value="xyz" Type="abc" Category="#"
                                Description="abc" Unit="abc" />
                                …
                                <Actuator Index="#" Id="abc" >
                                        <Field Id="#" Name="abc" Value="xyz" Type="abc"
                                        Category="#" Description="abc" Unit="abc" />
                                        …
                                </Actuator>
                                …
                        </Acm>
                        …
                </Platform>
                …
        </ CommUnit >
        …
</Reply>[CR/LF]
```

Description

Field nodes can be available at any level: CommUnit, Platform, Acm, or Actuator

- CommUnit.Id: [string] Unique ID (includes serial number for KCU, audio name for KAI…) of the device.
- Platform.Index: [int] Platform number
- Acm.Index: [int] ACM index
- Acm.Id: [string] (optional) ACM serial number (if supported by hardware)

- Actuator.Index: [int] Actuator number (0-…)
- Actuator.Id: [string] (optional) Actuator unique ID (if supported by hardware)
- Field.Id: [int] Field type unique ID
- Field.Name: [string] Field description
- Field.Value: [string/int/float] Field value
- Field.Type: [string] Field value type (int, float, or string)
- Field.Category: [int] Field type category (0 = state, 1 = measure, 2 = alarm).
    - An alarm (category =2) is limited to 2 values: 0 (inactive) ou 1 (active).
- Field.Description: [string] (optional) Description of the current value when the field is a state (category = 0). This attribute is not available for other categories.
- Field.Unit: [string] (optional) Units of the current value when available and when the field is a measure (category = 1). This attribute is not available for other categories.

5. Set hardware parameter

This command changes the current configuration value of a specific device in the hardware itself. The hardware can be connected to another computer, and it will preserve its settings.

Request

<Request Cmd="ConfigureParameter" CommUnitId="xyz" PlatformIndex="#" FieldId="#" Value="xyz" ExtraValue="xyz" />[CR/LF]

Description

- Cmd: [string] Command name
- CommUnitId: [string] Communication unit of the requested device
- PlatformIndex: [int] Platform number of the requested device
- FieldId [int]: Parameter type to set. Fields that can be configured are:
    - Delay (5007): Delay [float] to be applied on the stream in ms. The range is 0 to 247.5 ms.
    - Spectrum (5008): Motion or vibration attenuation [float] in dB configured for playback. Negative values are vibration attenuation while positive values are motion attenuation. The range is -20 to +20, 0 being full experience (no attenuation). Not all values are available depending on the hardware so the closest value, within range, is always used.
    - Intensity Level (5009): Attenuation level [float] in dB configured for motion playback. Values range from 0 (full experience) to -30 (30dB attenuation). There is also a special value of -100 that is used for mute. Not all values are available depending on the hardware so the closest value, within range, is always used.
- Value: Parameter value to set.
- ExtraValue: Optional parameter that can be used with these fields:
    - Spectrum (5008) when ExtraValue is set to "32768", the value will be persistent on the device. This value will be applied after a reboot (ex: power cycle).
    - Intensity Level (5009) when ExtraValue is set to "32768", the value will be persistent on the device. This value will be applied after a reboot (ex: power cycle).

Answer

<Reply Cmd="ConfigureParameter" CommUnitId="xyz" PlatformIndex="#" FieldId="#" Value="xyz" ExtraValue="xyz" Error="0" />[CR/LF]

Standard error codes apply (see *Communication format (API)* section for details).

Specific error codes for this command:

- 1: Hardware doesn't support configurable parameters
- 2: Parameter cannot be set (ready-only field or unsupported field)
- 3: Fail to update value (ie: communication issue)
- 4: Invalid value (ie: value out of range)

6. Set software parameter

This command changes the current software configuration value (persistent, even after computer reboots). This configuration applies to the computer itself so if the hardware is connected to another computer, it will use the configuration of the other computer.

Request

<Request Cmd="SetSoftwareParameter" FieldId="#" Value="xyz" />[CR/LF]

Description

- Cmd: [string] Command name
- FieldId [int]: Parameter type to set. Fields that can be configured are:
    - Local Mode (5004): Activation mode [int] of the platform, potential values:
        - **Park (0) – Platform powered off**
        - Hold Center (1) – Platform moved to center position
        - Hold Low (2) – Platform moved to low position
        - **Enable (3) – Platform ready to play a motion stream**
    - Default Mode (6000): Activation mode [int] of the platform to be applied when Local mode is Enable and there is no motion stream to play, potential values:
        - Park (0) – Platform powered off
        - Hold Center (1) – Platform moved to center position
        - Hold Low (2) – Platform moved to low position
    - Intensity reset on stream (5070): Intensity reset based on the stream status, potential values:
        - **Disabled (0) – Intensity is not changed**
        - Next Stream (3) – Intensity reset (once) when the next stream starts
        - Reset Now (4) – Immediate intensity reset
        - All Streams (7) - Intensity reset every time a stream starts
- Value: Parameter value to set. When this attribute is not present, the configuration is cleared and it reverts to the default value.

Answer

<Reply Cmd="SetSoftwareParameter" FieldId="#" Value="xyz" Error="0" />[CR/LF]

Standard error codes apply (see *Communication format (API)* section for details).

Specific error codes for this command:

- 2: Parameter cannot be set (unsupported field)
- 4: Invalid value (ie: value out of range)

7. Get software parameter

This command retrieves the current software configuration value (persistent, even after computer reboots).

Request

<Request Cmd="GetSoftwareParameter" FieldId="#" />[CR/LF]

Description

- Cmd: [string] Command name
- FieldId [int] (optional): Parameter type to get (all parameters are returned if no specific field is requested). Fields that can be configured are:
    - Local Mode (5004): Activation mode [int] of the platform, potential values:
        - **Park (0) – Platform powered off**
        - Hold Center (1) – Platform moved to center position
        - Hold Low (2) – Platform moved to low position
        - **Enable (3) – Platform ready to play a motion stream**
    - Default Mode (6000): Activation mode [int] of the platform to be applied when Local mode is Enable and there is no motion stream to play, potential values:
        - Park (0) – Platform powered off
        - Hold Center (1) – Platform moved to center position
        - Hold Low (2) – Platform moved to low position
    - Intensity reset on stream (5070): Intensity reset based on the stream status, potential values:
        - **Disabled (0) – Intensity is not changed**
        - Next Stream (3) – Intensity reset (once) when the next stream starts
        - Reset Now (4) – Immediate intensity reset
        - All Streams (7) - Intensity reset every time a stream starts

Answer

<Reply Cmd="GetSoftwareParameter" FieldId="#" Error="0" >
<Field Id="#" Name="abc" Value="xyz" Type="abc" Category="#" Description="abc" Unit="abc" />
<Field Id="#" Name="abc" Value="xyz" Type="abc" Category="#" Description="abc" Unit="abc" />
…
</Reply>[CR/LF]

Standard error codes apply (see *Communication format (API)* section for details).

Specific error codes for this command:

- 9: Parameter cannot be retrieved (unsupported field)