

DESTO/DEBJE

Event types

Q:marketing Aktiengesellschaft

Düsseldorfer Straße 193

45481 Mülheim an der Ruhr

Telefon (02 08) 30 15 0

Telefax (02 08) 30 15 555

Internet: www.Qmarketing.de

dialog@Qmarketing.de

1 Event types	4
1.1 General event types	4
1.1.1 Echo event - <i>com.abb.ispf.event.echo</i>	4
1.1.2 Success - <i>com.abb.ispf.event.success</i>	4
1.1.3 Error - <i>com.abb.ispf.event.error</i>	4
1.1.4 Device info - <i>com.abb.ispf.event.deviceinfo</i>	4
1.1.5 Discovery - <i>com.abb.ispf.event.discovery</i>	5
1.1.6 Delete event - <i>com.abb.ispf.event.delete</i>	5
1.1.7 Manage remote access	6
1.1.8 Delete client certificate - <i>com.abb.ispf.event.client-remove</i>	6
1.1.9 Push notification	6
1.2 Event types specific for KNX Security Panel	7
1.2.1 Alert - <i>com.abb.ispf.event.knx-security-panel.alert</i>	7
1.2.2 Command - <i>com.abb.ispf.event.knx-security-panel.command</i>	7
1.2.3 State - <i>com.abb.ispf.event.knx-security-panel.state</i>	7
1.2.4 Log - <i>com.abb.ispf.event.knx-security-panel.log</i>	7
1.3 Event types specific for Busch-Welcome	8
1.3.1 Pairing between App and Gateway - <i>com.abb.ispf.event.welcome.connect</i>	8
1.3.2 ACL-Request - <i>com.abb.ispf.event.welcome.acl-request</i>	8
1.3.3 ACL-Upload - <i>com.abb.ispf.event.welcome.acl-upload</i>	8
1.3.4 ACL-Fetch - <i>com.abb.ispf.event.welcome.acl-fetch</i>	9
1.3.5 Surveillance call - <i>com.abb.ispf.event.welcome.call-surveillance</i>	9
1.3.6 Ring - <i>com.abb.ispf.event.welcome.ring</i>	10
1.3.7 Missed call - <i>com.abb.ispf.event.welcome.call-missed</i>	10
1.3.8 Call answered - <i>com.abb.ispf.event.welcome.call-answered</i>	10

1.3.9 Hang up - <i>com.abb.ispf.event.welcome.call-terminated</i>	11
1.3.10 Screenshot - <i>com.abb.ispf.event.welcome.screenshot</i>	11
1.3.11 Switch light - <i>com.abb.ispf.event.welcome.light</i>	12
1.3.12 Open door - <i>com.abb.ispf.event.welcome.door-open</i>	12

1 Event types

The following event types will be available:

1.1 General event types

1.1.1 Echo event – *com.abb.ispf.event.echo*

Event can be sent bidirectional.

Used to test the websocket connection. The websocket server of the portal will send a message back to the portal client, containing the payload data, the portal client sent to the portal server.

1.1.2 Success – *com.abb.ispf.event.success*

Event can only be sent by the portal server.

Used to acknowledge the reception of an event from a websocket portal client.

1.1.3 Error – *com.abb.ispf.event.error*

Event can only be sent by the portal server.

Used to inform a portal client about an error that occurred while processing the last portal client's message.

1.1.4 Device info – *com.abb.ispf.event.deviceinfo*

Event can only be sent by portal client.

Used to inform the portal server about:

- Hardware version (not applicable for apps)
- Software version
- Current system time
- Feature set (only applicable for apps, e.g. "Can the camera be used?")

Example of payload:

```
{
  "hardware-version": "1.0",
  "software-version": "1.0",
  "timestamp": "2008-02-01T09:00:22+05:00",
  "feature-set": {
    "camera": true
  }
}
```

Further information about the feature-set should be provided by DEBJE.

1.1.5 Discovery – *com.abb.ispf.event.discovery*

Event can be sent bidirectional.

If the portal client sends a message of this type, the payload has to be omitted.

Used to request a list of addressable portal clients.

The portal server will respond with a event of type *com.abb.ispf.event.broadcast*, whereas the payload will be a list of addressable portal clients. The list will contain also the friendly name and type of the addressable portal clients.

Example of the payload:

```
{
  "84E7DFF7-44D0-48C2-80A9-99DB7C27CD86": { "name": "IP-Gateway, Loft",
"type": "com.abb.ispf.client.welcome.gateway" },
  "ADFB584B-BB24-46EA-A1A3-D39AE1A5F785": { "name": "IP-Gateway, Summer
cottage", "type": "com.abb.ispf.client.welcome.gateway" },
  "AAF7B4BB-AECA-4420-A875-893947FEDD22": { "name": null, "type":
"com.abb.ispf.client.welcome.gateway" }
}
```

1.1.6 Delete event – *com.abb.ispf.event.delete*

Event can only be sent by portal clients.

Used to inform the portal server about an event that should be deleted.

Example of event message:

```
{
  "id": "7C65AA03-8AD7-4213-94A7-39B0CAB8714A",
  "timestamp": "2008-02-01T09:00:22+05:00",
  "type": "com.abb.ispf.event.delete",
  "belongsto": "ADFB584B-BB24-46EA-A1A3-D39AE1A5F785"
}
```

To delete a specific event, the UUID of the event must be put into the field „belongsto“.

1.1.7 Manage remote access

To BE DISCUSSED:

We should not rely on the possibility to force the portal client to stop connecting to the portal by sending him a message (via HTTP-Long-Polling, Push-Notifications, Websocket, ...). The app may be closed, or the push notifications may be temporarily disabled. In this case, the information of a to be disabled connection may be lost.

To (temporarily) disable the remote access of a portal client, the certificate of the portal client should be added to the revocation list of the portal. Additionally, the SIP-server should provide the portal with an interface, where the updated revocation list can be pushed to. From our perspective, this is the only secure way to handle to prevent further connections from a stolen mobile device.

1.1.8 Delete client certificate – *com.abb.ispf.event.client-remove*

Event can only be sent by the portal server.

Used to request the portal client to delete its certificate and to no longer connect to the portal.

When the portal client receives a event of this type, the client must visit the HTTP API endpoint for certificate revocation, which is further specified in document „Certificate revocation“.

1.1.9 Push notification

To BE DISCUSSED:

This must be discussed again.

We think, that a single event for sending push notifications is not needed. Instead, the portal should have the knowledge, on which event occurrences it should push notifications to portal apps (e.g. Busch-Welcome Ring, KNX-SecurityPanel Alarm).

1.2 Event types specific for KNX Security Panel

1.2.1 Alert – *com.abb.ispf.event.knx-security-panel.alert*

Event can only be sent by portal client (KNX Security Panel).

Used to inform portal and apps about active alarms/faults in the KNX Security Panel.

1.2.2 Command – *com.abb.ispf.event.knx-security-panel.command*

From KNX Security Panel apps to portal server:

Used to send a command request to the portal server (setting, unsetting, reset, ...).

The KNX Security Panel app must specify the destination of the event (UUID of the KNX Security Panel portal client).

From portal server to KNX Security Panel:

Used to transmit command request to the KNX Security Panel.

1.2.3 State – *com.abb.ispf.event.knx-security-panel.state*

From KNX Security Panel:

Used to send informations about the current state of the KNX Security Panel to the portal. The portal will store this information for later transmission to KNX Security Panel apps.

From KNX Security Panel apps:

Used to retrieve informations about the current state of the KNX Security Panel from the portal.

1.2.4 Log – *com.abb.ispf.event.knx-security-panel.log*

From KNX Security Panel:

Used to send event logs of the KNX Security Panel to the portal server. The portal server will store this information for later transmission to the KNX Security Panel apps.

From KNX Security Panel apps:

Used to retrieve log entries of the KNX Security Panel.

1.3 Event types specific for Busch-Welcome

1.3.1 Pairing between App and Gateway – *com.abb.ispf.event.welcome.connect*

Event can only be sent by Busch-Welcome apps.

Used to send a request for authorization from App to an IP-Gateway.

The Busch-Welcome app must specify the destination IP-Gateway within the “destination” field of the event format.

Example for a complete event message:

```
{
  "id": "7C65AA03-8AD7-4213-94A7-39B0CAB8714A",
  "timestamp": "2008-02-01T09:00:22+05:00",
  "type": "com.abb.ispf.event.welcome.connect",
  "destination": [ "ADFB584B-BB24-46EA-A1A3-D39AE1A5F785" ]
}
```

1.3.2 ACL-Request – *com.abb.ispf.event.welcome.acl-request*

Event can only be sent by the portal server. The recipient will be the Busch-Welcome IP-Gateway.

Used to inform the Busch-Welcome IP-Gateway about an app, that wants to get access.

Example of the payload data:

```
{
  "uuid": "193CC63E-B21C-43F5-981D-D5BB32CFE3AA",
  "serial": "3127491",
  "name": "John Doe's iPhone"
}
```

1.3.3 ACL-Upload – *com.abb.ispf.event.welcome.acl-upload*

Event can only be sent by the portal client (Busch-Welcome IP-Gateway).

Used to inform the portal server about changes in the access control list of the IP-Gateway.

1.3.4 ACL-Fetch – *com.abb.ispf.event.welcome.acl-fetch*

Event can only be sent by the portal client (Busch-Welcome apps).

Used to fetch a list of privileges, a portal client (Welcome-App) has on one specific IP-Gateway.

1.3.5 Surveillance call – *com.abb.ispf.event.welcome.call-surveillance*

Portal client (Busch-Welcome IP-Gateway) to portal server.

Used to inform the portal server about a started surveillance call. The event must include which client started the surveillance call.

TO BE DISCUSSED:

Is it required, that the portal server will also be informed, if a surveillance call was started from a local client (Comfort-Panel, indoor audio/video station, locally connected mobile device)?

If this is the case, we propose the following payload data for the event:

```
{
  "client_local": "Friendly name of client - specified within the IP-
Gateway?"
}
```

If the client who started the surveillance call is a remote client (ISPF portal client, e.g. remotely connected Busch-Welcome app), the IP-Gateway instead should include the UUID of the portal client:

```
{
  "client_remote": "UUID of portal client"
}
```

Portal server to portal client (Busch-Welcome apps):

Used to inform the portal client a started surveillance call.

1.3.6 Ring – *com.abb.ispf.event.welcome.ring*

Portal client (Busch-Welcome IP-Gateway) to portal server:

Used to notify the portal server about a door ring.

Portal server to portal client (Busch-Welcome apps):

Used to inform the portal client about a door ring.

1.3.7 Missed call – *com.abb.ispf.event.welcome.call-missed*

Portal client (Busch-Welcome IP-Gateway) to portal server:

Used to notify the portal server about a missed call.

Portal server to portal client (Busch-Welcome apps):

Used to inform the portal client about a missed call.

1.3.8 Call answered – *com.abb.ispf.event.welcome.call-answered*

Portal client (Busch-Welcome IP-Gateway) to portal server:

Used to inform the portal server, that a door call has been answered. The portal client should specify in the payload, by which remote station (Comfort-Panel, indoor audio/video station, remote mobile device, local mobile device, ...) the call has been answered.

If the call was answered by a ISPF portal client, the ID/UUID of the portal client should be embedded within the payload data.

Proposal for payload data, if call was answered locally:

```
{
  "recipient_local": "Friendly name of client - specified within the IP-
Gateway?"
}
```

Proposal for payload data, if call was answered remotely:

```
{
  "recipient_remote": "UUID of portal client"
}
```

The event must include a reference (belongsto) to the previously occurred door ring event (*com.abb.ispf.event.welcome.ring*).

Portal server to portal client (Busch-Welcome apps):

Used to inform the portal client about an answered call.

1.3.9 Hang up – *com.abb.ispf.event.welcome.call-terminated*

Portal client (Busch-Welcome IP-Gateway) to portal server:

Used to inform the portal server about a terminated call.

The event must include a reference (belongsto) to the previously occurred event (com.abb.ispf.event.welcome.call-answered, com.abb.ispf.event.welcome.call-surveillance).

Portal server to portal client (Busch-Welcome apps):

Used to inform the portal client about a terminated call.

1.3.10 Screenshot – *com.abb.ispf.event.welcome.screenshot*

Portal client (Busch-Welcome IP-Gateway) to portal server:

Used to push a new screenshot to the portal server.

The event must include a reference (belongsto) to a previously occurred event:

- ID of ring event (com.abb.ispf.event.welcome.ring) for automatically taken screenshots
- ID of answered call event (com.abb.ispf.event.welcome.call-answered) for manually taken screenshots during call
- ID of surveillance call event (com.abb.ispf.event.welcome.call-surveillance) for manually taken screenshots during call

Portal server to portal client (Busch-Welcome apps):

Used to transmit a screenshot to a portal client.

1.3.11 Switch light – *com.abb.ispf.event.welcome.light*

Portal client (Busch-Welcome IP-Gateway) to portal server:

Used to inform the portal server that the light has been switched on or off.

We propose the following payload data, if the light has been switched on:

```
{
  "status": true
}
```

We propose the following payload data, if the light has been switched off:

```
{
  "status": false
}
```

Portal server to portal client (Busch-Welcome apps):

Used to inform the portal client that the light has been switched.

1.3.12 Open door – *com.abb.ispf.event.welcome.door-open*

Portal client (Busch-Welcome IP-Gateway) to portal server:

Used to inform the portal server, that the door has been opened.

The event should include from which client the door has been opened (Comfort-Panel, indoor audio/video station, locally connected mobile device, remotely connected mobile device).

Therefore, we propose the following payload data, if the client is locally connected to the IP-Gateway:

```
{
  "client_local": "Friendly name of client - specified within the IP-
Gateway?"
}
```

If the client is remotely connected, we propose the following payload data:

```
{
  "client_remote": "UUID of portal client"
}
```

To BE DISCUSSED:

Is it possible to open the door via mobile devices without an active call? If not, we propose, that the event should include a reference (belongsto) to a previously event (com.abb.ispf.event.welcome.call-surveillance or com.abb.ispf.event.welcome.call-answered).