



2N[®] Helios IP Event Log

IP Intercom



Configuration Manual

Firmware 2.11
Version 2.11

www.2n.cz

The 2N TELEKOMUNIKACE a.s. is a Czech manufacturer and supplier of telecommunications equipment.



The product family developed by 2N TELEKOMUNIKACE a.s. includes GSM gateways, private branch exchanges (PBX), and door and lift communicators. 2N TELEKOMUNIKACE a.s. has been ranked among the Czech top companies for years and represented a symbol of stability and prosperity on the telecommunications market for almost two decades. At present, we export our products into over 120 countries worldwide and have exclusive distributors on all continents.



2N[®] is a registered trademark of 2N TELEKOMUNIKACE a.s. Any product and/or other names mentioned herein are registered trademarks and/or trademarks or brands protected by law.



2N TELEKOMUNIKACE a.s. administers the FAQ database to help you quickly find information and to answer your questions about 2N products and services. On www.faq.2n.cz you can find information regarding products adjustment and instructions for optimum use and procedures „What to do if...“.



2N TELEKOMUNIKACE a.s. hereby declares that the 2N[®] Helios IP Event Log product complies with all basic requirements and other relevant provisions of the 1999/5/EC directive. For the full wording of the Declaration of Conformity see the CD-ROM (if enclosed) or our website at www.2n.cz.



This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.




The 2N TELEKOMUNIKACE a.s. is the holder of the ISO 9001:2009 certificate. All development, production and distribution processes of the company are managed by this standard and guarantee a high quality, technical level and professional aspect of all our products.


Content

1. Terms and Symbols	4
2. HTTP API General Description	5
3. Event Logging	8
4. HTTP API Message List	12
5. Logged Events	24
6. Format of Selected Parameters	29


1. Terms and Symbols


The following symbols and pictograms are used in the manual:

-  **Safety**
- **Always** abide by this information to prevent persons from injury.

-  **Warning**
- **Always** abide by this information to prevent damage to the device.

-  **Caution**
- **Important information** for system functionality.

-  **Tip**
- **Useful information** for quick and efficient functionality.

-  **Note**
- Routines or advice for efficient use of the device.

2. HTTP API General Description

2N[®] Helios IP HTTP API (Application Programming Interface) is an interface designed for **2N[®] Helios IP** remote control, configuration and data retrieval via third party applications. **2N[®] Helios IP** API enables 2N intercoms to be integrated in extensive user infrastructures (home automation, attendance control systems, etc.).

2N[®] Helios IP HTTP API provides a simple interface based on standard protocols (SOAP/HTTP), which helps read out state or statistic information.

2N[®] Helios IP HTTP API is defined through a web service available at:

- http://helios_ip_address/automation - **2N[®] Helios IP** control commands;
- http://helios_ip_address/notification - log data retrieval commands.

Send an HTTP request including a properly formatted SOAP/XML data to one of the above mentioned addresses. **2N[®] Helios IP** executes the requested operation and sends back a SOAP/XML response.

The general structure of the HTTP request is as follows:

```
POST /... HTTP/1.1
Host: ...
Content-Type: application/soap+xml
Content-Length: ...

<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Header>
    additional control information of request
  </s:Header>

  <s:Body>
    request parameters
  </s:Body>

</s:Envelope>
```

If the request is processed successfully, **2N[®] Helios IP** sends the following HTTP response:

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml
Content-Length: nnn

<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Header>
    additional control information of response
  </s:Header>

  <s:Body>
    Response parameters
  </s:Body>

</s:Envelope>
```

If the request cannot be processed or an error occurs during processing, **2N[®] Helios IP** sends the 400 Bad Request reply including SOAP/XML data with error identification.

```
HTTP/1.1 400 Bad Request
Content-Length: ...
Content-Type: application/soap+xml
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ter="http://www.onvif.org/ver10/error">
  <s:Body>
    <s:Fault>
      <s:Code>
        <s:Value> error code </s:Value>
        <s:Subcode>
          <s:Value> error subcode </s:Value>
        </s:Subcode>
      </s:Code>
      <s:Reason>
        <s:Text xml:lang="en"> error cause details </s:Text>
      </s:Reason>
      <s:Detail>
        <s:Text> additional error details </s:Text>
      </s:Detail>
    </s:Fault>
  </s:Body>
</s:Envelope>
```

3. Event Logging

2N® Helios IP implements an internal event log, which records events arising in the device – such as KeyPressed, CodeEntered, CardEntered, CallStateChanged and so on.

The log capacity is 500 events. When the log fills up, any new event deletes the oldest record. Events are arranged according to date and time (TimeStamp) in the log. The maximum time resolution is 1s. Also, each event is assigned a unique 16-bit RecordID, which is automatically incremented by 1s with every new event. The RecordID code helps distinguish between records of the same date and time.

The **2N® Helios IP** event logs can be arranged in a table:

	TimeStamp	RecordID	Event Type	Event Data
1	13/06/01 12:05:01	65533	CardEntered	Card=...
2	13/06/01 17:30:23	65534	CardEntered	Card=...
3	14/06/01 08:10:23	65535	KeyPressed	Key=...
4	14/06/01 08:10:23	0	KeyPressed	Key=..

500	27/06/01 17:05:01	496	CardEntered	Card=...

The first table row represents the oldest record (log beginning). The last row represents the most recent record (log end). Being a 16-bit number, the RecordID automatically overflows from 65535 to 0.

2N® Helios IP defines several types of events to be logged.

Event	Popis
KeyPressed	Keystroke
CallStateChanged	Call state change
CodeEntered	Numerical code entering
CardEntered	RFID card detection
DeviceState	Device state change

The **2N® Helios IP** event log readout is based on the WS-BaseNotification standard. There are two **2N® Helios IP** log event retrieval modes – **Subscription** and **PullPoint**.

In the **Subscription** mode, **2N[®] Helios IP** sends events to the subscriber in the form of Notify messages.

In the **PullPoint** mode, the client sends the GetMessages request to retrieve the log data.

Data Retrieval in Subscription Mode

To retrieve events from **2N[®] Helios IP** in the Subscription mode, subscribe via the **Subscribe** command, specifying the type of events to be retrieved, the HTTP sending address and the subscription time.

If the subscription is successful, **2N[®] Helios IP** sends the **SubscribeResponse** including the Subscription Id for subsequent communication with **2N[®] Helios IP**.

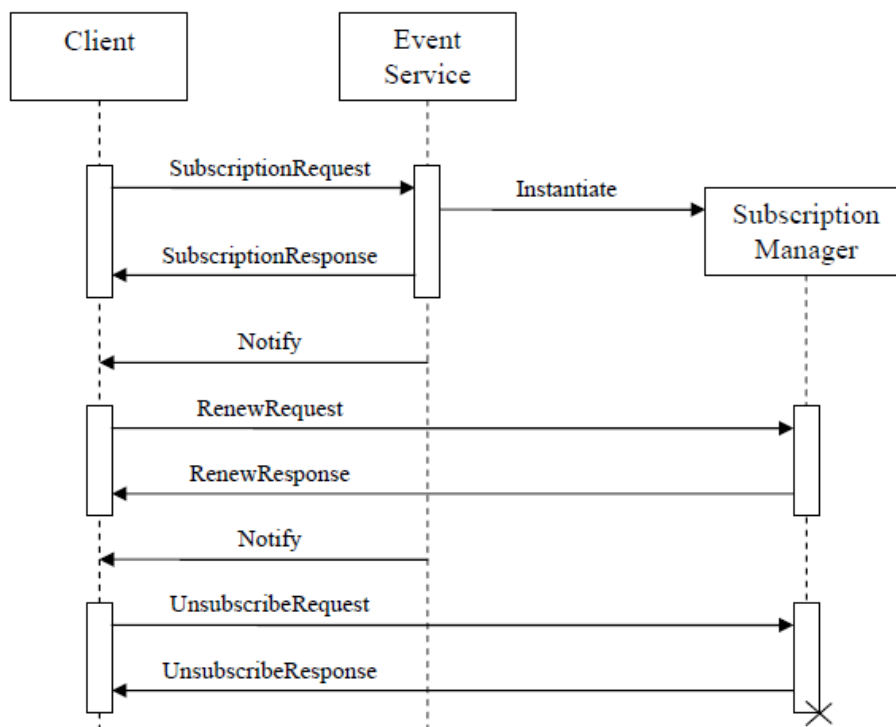
Now **2N[®] Helios IP** starts sending **Notify** messages to the specified HTTP address. Events are sent at the moment of rise without delay. In your Subscribe request, you can specify the maximum count of events to be sent within one Notify message.

The subscription time is limited (as specified in the Subscribe request). When the subscription time expires, you will be unsubscribed automatically. To extend the subscription time, send the **Renew** request before expiry. **2N[®] Helios IP** will reply with the **RenewResponse** message.

Send the **Unsubscribe** request to unsubscribe explicitly. **2N[®] Helios IP** will reply with the **UnsubscribeResponse** message.

If there is a firewall or NAT between you and **2N[®] Helios IP**, the Subscription mode may not be convenient. The Notify messages are delivered only if the network elements are properly configured.

The figure below shows the status diagram and typical client - **2N[®] Helios IP** message sequence.



Data Retrieval in PullPoint Mode

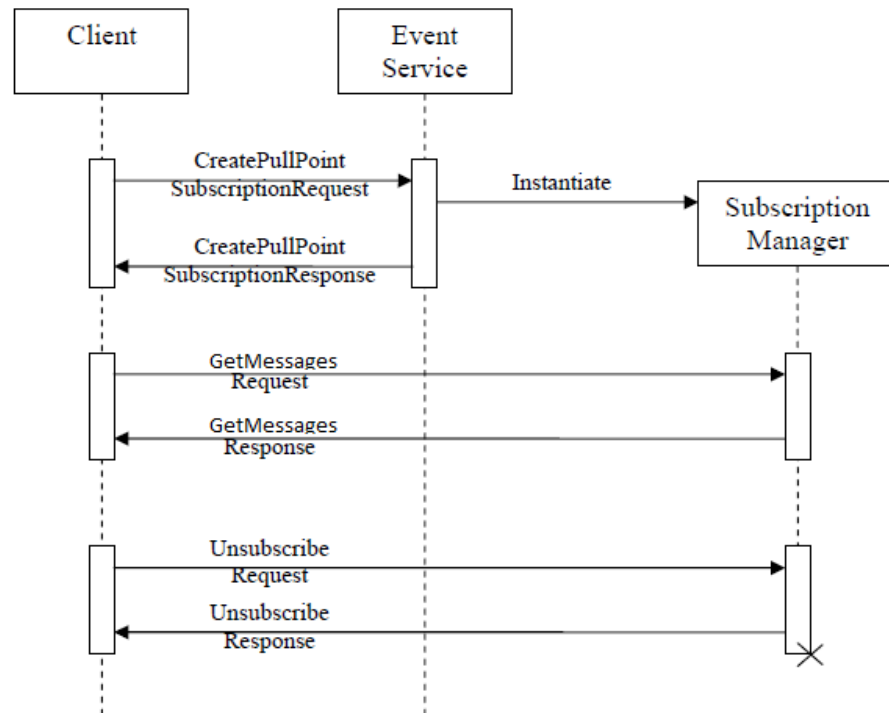
To retrieve events from **2N[®] Helios IP** in the PullPoint mode, create the PullPoint via the **CreatePullPoint** request, specifying the type of events to be retrieved and the subscription time.

If the subscription is successful, **2N[®] Helios IP** sends the **CreatePullPointResponse** including the Subscription Id for subsequent communication with **2N[®] Helios IP**.

Now you can read out events from **2N[®] Helios IP** periodically using the **GetMessages** request. **2N[®] Helios IP** sends the **GetMessagesResponse** including the list of events. In case there is no event in the queue, the event list is empty in the response.

The subscription time is limited (as specified in the CreatePullPoint request). When the subscription time expires, you will be unsubscribed automatically. To extend the subscription time, send the **Renew** request before expiry. **2N[®] Helios IP** will reply with the **RenewResponse** message.

Send the **Unsubscribe** request to unsubscribe explicitly. **2N[®] Helios IP** will reply with the **UnsubscribeResponse** message.



4. HTTP API Message List

2N® Helios HTTP API defines the following types of messages:

- [Subscribe](#)
- [SubscribeResponse](#)
- [CreatePullPoint](#)
- [CreatePullPointResponse](#)
- [Renew](#)
- [RenewResponse](#)
- [Unsubscribe](#)
- [UnsubscribeResponse](#)
- [GetMessages](#)
- [GetMessagesResponse](#)
- [Notify](#)

See below for details on the messages and their parameters and use.

Subscribe

The Subscribe request helps you subscribe to retrieval of **2N® Helios IP** events in the Subscribe mode, in which **2N® Helios IP** sends events autonomously in the form of Notify messages to the HTTP address specified in the request.

Send the Subscribe request to the **2N® Helios IP** address:
http://helios_ip_address/notification.

2N® Helios IP will reply with the SubscribeResponse message.

Subscribe request format:

```

<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
  xmlns:a="http://www.w3.org/2005/08/addressing"
  xmlns:event2n="http://www.2n.cz/2013/event">
  <s:Header>
  </s:Header>
  <s:Body>
    <wsnt:Subscribe>
      <wsnt:ConsumerReference>
        <a:Address>
          {HTTP_CONSUMER_ADDRESS}
        </a:Address>
      </wsnt:ConsumerReference>
      <wsnt:Filter>
        <wsnt:TopicExpression Dialect=
          "http://www.2n.cz/2013/TopicExpression/Multiple">
          {EVENTS_LIST}
        </wsnt:TopicExpression>
      </wsnt:Filter>
      <wsnt:InitialTerminationTime>
        {DATETIME_OR_DURATION}
      </wsnt:InitialTerminationTime>
      <wsnt:SubscriptionPolicy>
        <event2n:MaximumNumber>
          {MAX_NUMBER_OF_MSGS_AT_ONCE}
        </event2n:MaximumNumber>
        <event2n:SimpleMessages>
          {SIMPLE_MSGS_ENABLED_FLAG}
        </event2n:SimpleMessages>
        <event2n:StartRecordId>
          {START_RECORD_ID}
        </event2n:StartRecordId>
        <event2n:StartTimestamp>
          {START_TIMESTAMP}
        </event2n:StartTimestamp>
      </wsnt:SubscriptionPolicy>
    </wsnt:Subscribe>
  </s:Body>
</s:Envelope>

```

Message Parameters

HTTP_CONSUMER_ADDRESS

Specify the address to which the Notify messages including the requested events shall be sent. Keep the HTTP URI address format, e.g. `http://192.168.23.73/notify_server`. The parameter is obligatory.

DATETIME_OR_DURATION

Set the subscription time. Subscription will be cancelled automatically by when this time period expires. Send the Renew request before expiry to extend the subscription time. Enter the absolute time (DateTime) or relative duration (Duration) to specify the subscription time; refer to the Format of Selected Parameters subsection. The parameter is obligatory.

EVENTS_LIST

Make up a list of requested events. Separate the event types with a comma, e.g. event2n:KeyPressed, event2n:CardEntered. The parameter is optional. If you define no events, you will get all types of events.

MAX_NUMBER_OF_MSGS_AT_ONCE

Set the maximum count of events to be sent within one Notify message. The parameter is optional. If no value is completed, up to 10 events will be sent in one Notify message.

SIMPLE_MSGS_ENABLED_FLAG

Specify the format of event elements to be sent in the Notify message. If you set 0, the ONVIF standard (extended messages) will be applied. If you set 1, a more efficient format will be used (as described herein). The parameter is optional. If no value is defined, a more efficient format is used.

START_TIMESTAMP

Set the data and time for the oldest log event to be the first in the Notify message. The parameter is optional. If no value is set, all log events will be sent.

START_RECORD_ID

Set the RecordId for the log event to be the first in the Notify message. This parameter can only be used together with the START_TIMESTAMP parameter. The parameter is optional. If no value is set, all log events starting with the time specified in the START_TIMESTAMP parameter will be sent.

SubscribeResponse

2N® Helios IP sends the SubscribeResponse message in reply to the Subscribe request to inform the client that the client's subscription has been completed successfully.

```

<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa5="http://www.w3.org/2005/08/addressing"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
  xmlns:event2n="http://www.2n.cz/2013/event">
  <s:Header>
    <wsa5:Action s:mustUnderstand="true">
      http://docs.oasis-open.org/wsn/bw-2/NotificationProducer/SubscribeResponse
    </wsa5:Action>
  </s:Header>
  <s:Body>
    <wsnt:SubscribeResponse>
      <wsnt:SubscriptionReference>
        <wsa5:Address>
          {NOTIFICATIONS_PRODUCER_ADDRESS}
        </wsa5:Address>
        <wsa5:ReferenceParameters>
          <event2n:SubscriptionId>
            {SUBSCRIPTION_ID}
          </event2n:SubscriptionId>
        </wsa5:ReferenceParameters>
      </wsnt:SubscriptionReference>
      <wsnt:CurrentTime>
        {CURRENT_TIME}
      </wsnt:CurrentTime>
      <wsnt:TerminationTime>
        {EXPIRE_TIME}
      </wsnt:TerminationTime>
    </wsnt:SubscribeResponse>
  </s:Body>
</s:Envelope>

```

Message Parameters

NOTIFICATIONS_PRODUCER_ADDRESS

Specify the address for subsequent subscription communication with **2N® Helios IP** (i. e. a new address for Unsubscribe and Renew requests). In the present implementation, the address is `http://helios_ip_address/notification`.

SUBSCRIPTION_ID

A unique subscription identification code to be used for subsequent Renew and Unsubscribe communication with **2N® Helios**. Edit the Id as a common text.

CURRENT_TIME

Subscription date and time from the viewpoint of **2N® Helios IP**. Refer to the Format of Selected Parameters for format details.

EXPIRE_TIME

Specify the subscription date and time. Use the Renew request to extend the subscription time before expiry. Refer to the Format of Selected Parameters for format details.

CreatePullPoint

The CreatePullPoint request helps you subscribe to **2N® Helios IP** event retrieval in the PullPoint mode, in which you read out events from the log actively using the GetMessages request.

Send the CreatePullPoint request to the **2N® Helios IP** address:
http://helios_ip_address/notification.

2N® Helios IP will reply with the CreatePullPointResponse.

CreatePullPoint request format:

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
  xmlns:event2n="http://www.2n.cz/2013/event">
  <s:Header>
  </s:Header>
  <s:Body>
    <wsnt:CreatePullPoint>
      <wsnt:Filter>
        <wsnt:TopicExpression Dialect=
          "http://www.2n.cz/2013/TopicExpression/Multiple">
          {EVENTS_LIST}
        </wsnt:TopicExpression>
      </wsnt:Filter>
      <wsnt:InitialTerminationTime>
        {DATETIME_OR_DURATION}
      </wsnt:InitialTerminationTime>
      <wsnt:SubscriptionPolicy>
        <event2n:SimpleMessages>
          {SIMPLE_MSGS_ENABLED_FLAG}
        </event2n:SimpleMessages>
        <event2n:StartRecordId>
          {START_RECORD_ID}
        </event2n:StartRecordId>
        <event2n:StartTimestamp>
          {START_TIMESTAMP}
        </event2n:StartTimestamp>
      </wsnt:SubscriptionPolicy>
    </wsnt:CreatePullPoint>
  </s:Body>
</s:Envelope>
```


CreatePullPointResponse

2N® Helios IP sends the CreatePullPointResponse in reply to the CreatePullPoint request to inform the client that the client's subscription has been completed successfully.

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa5="http://www.w3.org/2005/08/addressing"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
  xmlns:event2n="http://www.2n.cz/2013/event">
  <s:Header>

  <wsa5:Action>//http://docs.oasis-open.org/wsn/bw-2/CreatePullPoint/CreatePullPointResponse
  </wsa5:Action>
  </s:Header>
  <s:Body>
    <wsnt:CreatePullPointResponse>
      <wsnt:SubscriptionReference>
        <wsa5:Address>
          {NOTIFICATIONS_PRODUCER_ADDRESS}
        </wsa5:Address>
        <wsa5:ReferenceParameters>
          <event2n:SubscriptionId>
            {SUBSCRIPTION_ID}
          </event2n:SubscriptionId>
        </wsa5:ReferenceParameters>
      </wsnt:SubscriptionReference>
      <wsnt:CurrentTime>
        {CURRENT_TIME}
      </wsnt:CurrentTime>
      <wsnt:TerminationTime>
        {EXPIRE_TIME}
      </wsnt:TerminationTime>
    </wsnt:CreatePullPointResponse>
  </s:Body>
</s:Envelope>
```

Message Parameters

NOTIFICATIONS_PRODUCER_ADDRESS

Specify the address for subsequent subscription communication with **2N® Helios IP** (i. e. a new address for Unsubscribe, Renew and GetMessages requests). In the present implementation, the address is `http://helios_ip_address/notification`.

SUBSCRIPTION_ID

A unique subscription identification code to be used for subsequent Renew, Unsubscribe and GetMessages communication with **2N® Helios**. Edit the Id as a common text.

CURRENT_TIME

Subscription date and time from the viewpoint of **2N® Helios IP**. Refer to the Format of Selected Parameters for format details.

EXPIRE_TIME

Specify the subscription date and time. Use the Renew request to extend the subscription time before expiry. Refer to the Format of Selected Parameters for format details.

Renew

The Renew request helps you extend your subscription time. Send the request to the address included in the SubscribeResponse message. **2N® Helios IP** replies with the RenewResponse message.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
  xmlns:event2n="http://www.2n.cz/2013/event">
  <s:Header>
    <event2n:SubscriptionId a:IsReferenceParameter="true">
      {SUBSCRIPTION_ID}
    </event2n:SubscriptionId>
  </s:Header>
  <s:Body>
    <wsnt:Renew>
      <wsnt:TerminationTime>
        {DATETIME_OR_DURATION}
      </wsnt:TerminationTime>
    </wsnt:Renew>
  </s:Body>
</s:Envelope>
```

Message Parameters

SUBSCRIPTION_ID

A unique subscription identification code included in the SubscriptionResponse. The parameter is obligatory.

DATETIME_OR_DURATION

Extend the subscription time. Subscription will be cancelled automatically by **2N® Helios IP** when this time period expires. Send another Renew request before expiry to extend the subscription time. Enter the absolute time (DateTime) or relative duration (Duration) to specify the subscription time; refer to the Format of Selected Parameters subsection. The parameter is obligatory.

RenewResponse

The RenewResponse message is a reply to the Renew request, in which **2N® Helios IP** informs the client of the prolongation of the client's subscription time.

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa5="http://www.w3.org/2005/08/addressing"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2">
  <s:Header>
    <wsa5:Action s:mustUnderstand="true">
      http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/RenewResponse
    </wsa5:Action>
  </s:Header>
  <s:Body>
    <wsnt:RenewResponse>
      <wsnt:TerminationTime>
        {CURRENT_TIME}
      </wsnt:TerminationTime>
      <wsnt:CurrentTime>
        {EXPIRE_TIME}
      </wsnt:CurrentTime>
    </wsnt:RenewResponse>
  </s:Body>
</s:Envelope>
```

Message Parameters

CURRENT_TIME

Subscription date and time from the viewpoint of **2N® Helios IP**. Refer to the Format of selected parameters for format details.

EXPIRE_TIME

Specify the subscription date and time. Use the Renew request to extend the subscription time before expiry. Refer to the Format of Selected Parameters for format details.

Unsubscribe

The Unsubscribe request helps you terminate your subscription. Send this request to the address included in the SubscribeResponse message. **2N® Helios IP** replies with the UnsubscribeResponse.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
  xmlns:event2n="http://www.2n.cz/2013/event">
  <s:Header>
    <event2n:SubscriptionId a:isReferenceParameter="true">
      {SUBSCRIPTION_ID}
    </event2n:SubscriptionId>
  </s:Header>
  <s:Body>
    <wsnt:Unsubscribe>
    </wsnt:Unsubscribe>
  </s:Body>
</s:Envelope>
```

Message Parameters

SUBSCRIPTION_ID

A unique subscription identification code included in the SubscriptionResponse. The parameter is obligatory.

UnsubscribeResponse

The UnsubscribeResponse message is a reply to the Unsubscribe request, in which **2N[®] Helios IP** confirms the client's subscription termination.

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa5="http://www.w3.org/2005/08/addressing"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2">
  <s:Header>
    <wsa5:Action s:mustUnderstand="true">
      http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/UnsubscribeResponse
    </wsa5:Action>
  </s:Header>
  <s:Body>
    <wsnt:UnsubscribeResponse>
    </wsnt:UnsubscribeResponse>
  </s:Body>
</s:Envelope>
```

GetMessages

The GetMessages request helps you read out events actively from **2N[®] Helios IP**. Send the request to the address included in the CreatePullPointResponse. **2N[®] Helios IP** replies with the GetMessageResponse.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
  xmlns:event2n="http://www.2n.cz/2013/event">
  <s:Header>
    <event2n:SubscriptionId a:isReferenceParameter="true">
      {SUBSCRIPTION_ID}
    </event2n:SubscriptionId>
  </s:Header>
  <s:Body>
    <wsnt:GetMessages>
      <wsnt:MaximumNumber>
        {MAXIMUM_MSGS_IN_RESPONSE}
      </wsnt:MaximumNumber>
    </wsnt:GetMessages>
  </s:Body>
</s:Envelope>
```

Message Parameters

SUBSCRIPTION_ID

A unique subscription identification code included in the SubscriptionResponse. The parameter is obligatory.

MAXIMUM_MSGS_IN_RESPONSE

Set the maximum count of events to be sent within the GetMessagesResponse. The parameter is optional. If no value is completed, up to 10 events will be sent in one GetMessagesResponse message.

GetMessagesResponse

The GetMessagesResponse message is a reply to the GetMessages request.

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:topic2n="http://www.2n.cz/2013/event/topics"
  xmlns:wsa5="http://www.w3.org/2005/08/addressing"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
  xmlns:event2n="http://www.2n.cz/2013/event">
  <s:Header>
    <wsa5:Action s:mustUnderstand="true">
      http://docs.oasis-open.org/wsn/bw-2/PullPoint/GetMessagesResponse
    </wsa5:Action>
  </s:Header>
  <s:Body>
    <wsnt:GetMessagesResponse>
      {EVENT_LIST}
    </wsnt:GetMessagesResponse>
  </s:Body>
</s:Envelope>
```

Message Parameters

EVENT_LIST

Refer to the Logged Events subsection for the list of events. The event list can be empty. Specify the maximum count of events to be sent at one time in the CreatePullPoint request in the Subscribe or the GetMessages request.

Notify

The Notify message informs you of the rise of a new event in the Subscription mode. **2N® Helios IP** sends Notify to the address specified in the Subscribe request. The Notify message have to be confirmed by the OK message from the opposite party. The next Notify message is sent after confirmation or after aprox 5 s timeout.

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:topic2n="http://www.2n.cz/2013/event/topics"
  xmlns:event2n="http://www.2n.cz/2013/event"
  xmlns:wsa5="http://www.w3.org/2005/08/addressing"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2">
  <s:Header>
    <wsa5:Action s:mustUnderstand="true">
      http://docs.oasis-open.org/wsn/bw-2/NotificationConsumer/Notify
    </wsa5:Action>
  </s:Header>
  <s:Body>
    <wsnt:Notify>
      {EVENT_LIST}
    </wsnt:Notify>
  </s:Body>
</s:Envelope>
```

Message Parameters

EVENT_LIST

Refer to the Logged Events subsection for the list of events. The message always includes one event at least. Specify the maximum count of events in the Subscribe request.

5. Logged Events

This subsection includes a list of events to be stored in the **2N[®] Helios IP** log and specifies the transfer format via the Notify and GetMessagesResponse messages.

- [KeyPressed](#)
- [CallStateChanged](#)
- [CardEntered](#)
- [CodeEntered](#)
- [DeviceState](#)

See below for details on the events and their parameters and use.

KeyPressed

The KeyPressed event is generated whenever a key is pressed on the device.

```
<event2n:Msg>
<event2n:Id> {RECORD_ID} </event2n:Id>
<event2n:Timestamp> {TIMESTAMP} </event2n:Timestamp>
<event2n:EventName>event2n:KeyPressed</event2n:EventName>
<event2n>Data>
  <event2n:Key> {KEYCODE} </event2n:Key>
</event2n>Data>
</event2n:Msg>
```

Event Parameters

RECORD_ID

Event serial number.

TIMESTAMP

Event date and time; refer to the Format of Selected Parameters.

KEYCODE

Code of the key pressed. Applicable values:

- 0, 1, ..., 9, *, # – numerical keypad buttons
- %1..%54 – call buttons

CallStateChanged

The CallStateChanged event is generated whenever a change of the currently made call occurs.

```

<event2n:Msg>
  <event2n:Id>{RECORD_ID}</event2n:Id>
  <event2n:Timestamp>{TIMESTAMP}</event2n:Timestamp>
  <event2n:EventName>event2n:CallStateChanged</event2n:EventName>
  <event2n:Data>
    <event2n:State>{STATE}</event2n:State>
    <event2n:Direction>{DIRECTION}</event2n:Direction>
  </event2n:Data>
</event2n:Msg>

```

Event Parameters

RECORD_ID

Event serial number.

TIMESTAMP

Event date and time; refer to the Format of Selected Parameters.

STATE

Event type specification. Applicable values:

- **ringing** – ringing
- **connected** – call connected
- **terminated** – call terminated

DIRECTION

Call direction. Applicable values:

- **incoming** – for incoming calls
- **outgoing** – for outgoing calls

CardEntered

The CardEntered event is generated whenever a card is applied to the RFID card reader connected to the device.

```
<event2n:Msg>
  <event2n:Id>{RECORD_ID}</event2n:Id>
  <event2n:Timestamp>{TIMESTAMP}</event2n:Timestamp>
  <event2n:EventName>event2n:CardEntered</event2n:EventName>
  <event2n:Data>
    <event2n:Card>{CARD}</event2n:Card>
    <event2n:Valid>{VALID_FLAG}</event2n:Valid>
    <event2n:Reader>{READER_ID}</event2n:Reader>
    <event2n:Direction>{DIRECTION}</event2n:Direction>
  </event2n:Data>
</event2n:Msg>
```

Event Parameters

RECORD_ID

Event serial number.

TIMESTAMP

Event date and time; refer to the Format of Selected Parameters.

CARD

Detected card identification with a 24-64-bit number in the hexadecimal format, e.g. 54A86B87C5.

VALID_FLAG

If the detected card is valid, the parameter value is **true**. If not, the parameter is set to **false**.

READER_ID

Card reader identification. Applicable values:

- **internal_cardreader** - internal card reader at **2N[®] Helios IP Vario, 2N[®] Helios IP Force**.
- **external_cardreader** - external card reader connected via Wiegand interface at **2N[®] Helios IP Vario, 2N[®] Helios IP Force**.
- **Jméno modulu** - for **2N[®] Helios IP Verso** card reader ID is configured in menu Hardware / Extenders / Modules, parameter Module Name for used card reader.

DIRECTION

Inteded direction of a person configured in an intercom. Applicable values:

- **In** – for incoming people.
- **Out** – for outgoing people.
- **Unspecified** – not specified.

CodeEntered

The CodeEntered event is generated whenever a numerical code is entered and confirmed via the device keypad.

```
<event2n:Msg>
  <event2n:Id>{RECORD_ID}</event2n:Id>
  <event2n:Timestamp>{TIMESTAMP}</event2n:Timestamp>
  <event2n:EventName>event2n:CodeEntered</event2n:EventName>
  <event2n:Data>
    <event2n:Code> {CODE}</event2n:Code>
    <event2n:Valid> {VALID_FLAG}</event2n:Valid>
  </event2n:Data>
</event2n:Msg>
```

Event Parameters

RECORD_ID

Event serial number.

TIMESTAMP

Event date and time; refer to the Format of Selected Parameters.

CODE

Numerical code entered, e.g. 1234.

VALID_FLAG

If the entered code is valid (matches the **2N® Helios IP** valid codes), the parameter value is **true**. If not, the parameter is set to **false**.

DeviceState

The DeviceState event is generated whenever any of the selected device state changes occurs.

```
<event2n:Msg>
  <event2n:Id>{RECORD_ID}</event2n:Id>
  <event2n:Timestamp>{TIMESTAMP}</event2n:Timestamp>
  <event2n:EventName>event2n:DeviceState</event2n:EventName>
  <event2n:Data>
    <event2n:State>{STATE}</event2n:State>
  </event2n:Data>
</event2n:Msg>
```

Event Parameters

RECORD_ID

Event serial number.

TIMESTAMP

Event date and time; refer to the Format of Selected Parameters.

STATE

Event type specification. Applicable values:

- **startup** – event generated upon device start

6. Format of Selected Parameters

TimeStamp – Absolute Time

The absolute time parameter obeys the ISO8601 subset standard: it is set as a string of characters in the CCYY-MM-DDThh:mm:ss format, where CC stands for the century, YY for the year, MM for the month and DD for the day. The T character separates date and time. hh represents hours, mm minutes and ss seconds.

Examples:

- 2013-05-31T13:20:00 – 31.5.2013 13:20

Duration

The duration parameter obeys the extended ISO8601 standard: it is set as a string of characters in the PnYnMnDTnHnMnS format, where nY stands for the year count, nM for the month count and nD for the day count. T is the date/time separator, nH gives the count of hours, nM gives the count of minutes and nS sets the count of seconds. n must be an integer.

Examples:

- PT60S 60 seconds
- P1DT12H 1 day and 12 hours

**2N TELEKOMUNIKACE a.s.**

Modřanská 621, 143 01 Prague 4, Czech Republic
Phone: +420 261 301 500, Fax: +420 261 301 599
E-mail: sales@2n.cz
Web: www.2n.cz

2043v1