

BS1000 command and backlog protocol

Introduction

When the bs1000 is updating a website, measurement data is transferred to the site using a http request. However, there are occasions when this transfer fails, usually because the network is down.

The measurement data is saved in the on-board dataflash memory, but remains there. No action is started from the bs1000 to retransmit the data.

Either the server has to open a connection back to the BS1000 to fetch the data using the data.xml page of the BS1000, or it has to report to the station it is missing some data and request a retransmit.

Since firmware version 1.0.4.7 it is possible to use the last option. This option does not require a connection back to the station. It does require some modifications on the server side in order to use this protocol.

The second part of this document provides information on how the same mechanism is used to obtain general information from the bs1000 remotely, without opening a direct connection. (v1.0.5.2)

Sequence

The following sequence should be performed in order to retransmit data:

1. BS1000 transmits a http request to the server with measurement data.
2. The server responds to the request with a normal empty response (no further action performed by the BS1000), or it responds with a small xml file.
3. The BS1000 interprets the xml response and if successful, it will transmit the data specified in the xml as a new http-POST request.

The key element in this sequence is that the basestation interprets the returned xml, and performs the action described there.

XML: command

When action of the bs1000 is needed then an xml should be sent as a reply with root element 'cmd'. The 'cmd' element has the following attributes:

Attribute	Type	Function
cmd	int	Command identifier (1 for backlog)
url	string	The location where to report back
id	int	Filter data by this Sensor id
q	int	Filter data by this Sensor type
s	int	Filter data by starting from this Start time
e	int	Filter data by ending to this End time
site	int	Site id to be used on report back
log	string	Logger id to be used on report back

pwd	string	Password to be used on report back
------------	--------	------------------------------------

With this data, the basestation will first select data from its dataflash memory. Only measurements within the [s,e] period are reported, from the sensor specified.

```
<?xml version="1.0" encoding="utf-8"?>
<cmd cmd="1"
  url= "add_backlog.php"
  id="16408"
  q="1"
  s="422081202"
  e="422123724"
  site="183"
  pwd="some pwd"
  log="log93">
</cmd>
```

Example of a command sent as response to an incoming measurement. It requests data from temperature sensor 16408 for the period s, e. The site, pwd and log attributes are used in the new http request from the bs1000.

This selection is sent to the specified url as a http-POST request, using the **site**, **log** and **pwd** fields as parameters. An extra parameter '**data**' is sent as well, with the selected measurements from the dataflash memory in xml format. The parameters **e**, **id** and **type** are reflected as well.

XML: response data

The format of xml data reported back is the same as for the data.xml page:

The data consists of the 'measurements' root element, which contains a list of 0, 1 or more value elements. Each value element represent a measurement, and it has the following attributes:

- **id** sensor id
- **type** sensor type; 1=temperature, 3= RH%, 5=CO2, ...
- **t** timestamp; number of seconds since 1-1-2000

The measurement value itself is given as the element value.

```
<?xml version="1.0" encoding="utf-8" ?>
<measurements>
<value id="16408" type="1" t="327682224">25.1</value>
<value id="16408" type="1" t="327682287">39.5</value>
<value id="16408" type="1" t="327682335">33.7</value>
<value id="16408" type="1" t="327682378">30.7</value>
<value id="16408" type="1" t="327682417">28.9</value>
<value id="16408" type="1" t="327682493">26.8</value>
<value id="16408" type="1" t="327682581">25.4</value>
<value id="16408" type="1" t="327682660">24.7</value>
<value id="16408" type="1" t="327682718">24.4</value>
<more>327684165</more>
</measurements>
```

Example of xml data reported back to the server.

Since the data transfer could potentially take a long time, the resulting xml data may be truncated. This is indicated by the 'more' element. The element value of the 'more' element gives the last timestamp that was examined in the flash memory. Note that this may not be the same as the last given value timestamp. When the 'more' element is not present, the data is not truncated.

Firmware

Typically about 10 measurements are reported back per request. Although the cmd – response pair could be repeated continuously (the website creates another 'cmd' based on the response), the firmware is a limiting factor at this moment: When a command initiates a new request, then during this time the firmware blocks other network requests, like sending http-requests from rules. This means outgoing measurement requests are blocked during backlog activity. Therefore it is advised not to continue backlog, but initiate a backlog only after a measurement is reported.

In the next firmware a timed rule will be defined, that will generate a http request based on a timer, not on an incoming measurement. This request can trigger a backlog operation, or some other commands yet to be specified.

Remote commands

Since firmware version 1.0.5.2 an extra rule type is defined that will generate a http-request based on a timer, not on an incoming measurement. This request can trigger a backlog operation, or an action specified by command:

Command	Action
1	Provide a backlog
2	Provide general information
3	Provide recent event log
4	Set time

The command is an xml file as response to a http request. The timed rule is fired at most every 15 seconds. The inhibit time specified in the rule is added to this period.

Command 1 is specified as above, the commands 2-4 have the following attributes:

Attribute	Type	Function
cmd	int	Command identifier
url	string	The location where to report back
site	int	Site id to be used on report back
id	int	Command id to be used on report back
log	string	Logger id to be used on report back
pwd	string	Password to be used on report back

s	int	Time to set for command 4
---	-----	---------------------------

Again, the site, log, id and pwd data fields are reflected on the response http-request.

Command 2: Provide general information

With command 2 general information about the logger is sent back:

A http-request is sent to the provided url, together with the site, log and pwd data fields and the data parameter that provides the following xml:

The data consists of the 'info' root element, and has the following child elements:

Element	Element	type	Description
device			
	model	string	Device model name
	name	string	Network name
	version	string	Firmware version
	utc_time	string	Current time of internal clock (UTC)
	utc_date	string	Current date of internal clock (UTC)
network			
	mac	string	ethernet identification
	dhcp	yes / no	indicates if dhcp is used
	ip	string	ip address of bs1000
	mask	string	network mask
	gateway	string	gateway ip address
	dns	string	dns server ip address
	time	string	time server ip address

Example

The server responds to a http-post request from the bs1000 with this xml to issue command 2:

```
<cmd cmd="2"
url="192.168.0.62/temperature/cmd_result.php"
site="138"
id="56"
log="logger"
pwd="password">
</cmd>
```

Then the bs1000 will send a new http-post request to the server as indicated in the url field with these parameters as data:

```

id = "56"
site="138"
pwd="password"
log="logger"
data = "<?xml version="1.0" encoding=utf-8"?>
<info>
<device>
<model>BS1000</model>
<name>log53</name>
...
</log>"

```

Command 3: Provide the recent event log

Command 3 lets the logger send back the current recent event log.

A http-request is sent to the provided url, together with the site, log and pwd data fields and the data parameter that provides the following xml:

The data consists of the 'log' root element, and has the following child elements:

Element	Attribute	type	Description
event		int	Event result number
	t	string	UTC date / time
	r	int	Rule number
	d	int	Device number

Example

The server responds to a http-post request from the bs1000 with this xml to issue command 3:

```

<cmd cmd="3"
url="192.168.0.62/temperature/cmd_result.php"
site="138"
id="57"
log="logger"
pwd="password">
</cmd>

```

Then the bs1000 will send a new http-post request to the server with these parameters as data:

```

id = "57"
site="138"
pwd="password"
log="logger"
data = "<?xml version="1.0" encoding=utf-8"?>
<log>
<event r="1" d="56" t="2013-05-31 11:29:17">200</event>
...

```

```
</log>".
```

Command 4: Set time

This command enables you to set the loggers clock to a specified time.

The cmd element additionally consists of the attribute 's' that contains the UTC time to be set, in seconds since 1-1-2000.

The bs1000 responds to this command by setting the internal clock, and emit the following xml:

Element	Element	type	Description
time			
	utc_date	string	UTC time
	utc_time	string	UTC date

Example

The server responds to a http-post request from the bs1000 with this xml to issue command 4:

```
<cmd cmd="3"
url="192.168.0.62/temperature/cmd_result.php"
site="138"
id="58"
log="logger"
pwd="password"
s="423315019">
</cmd>
```

Then the bs1000 will set the clock to 423315019 and send a new http-post request to the server with these parameters as data:

```
id = "58"
site="138"
pwd="password"
log="logger"
data = "<?xml version="1.0" encoding=utf-8"?>
<time>
<utc_time>11:30:19</utc_time>
<utc_date>2013-05-31</utc_date>
</time>".
```