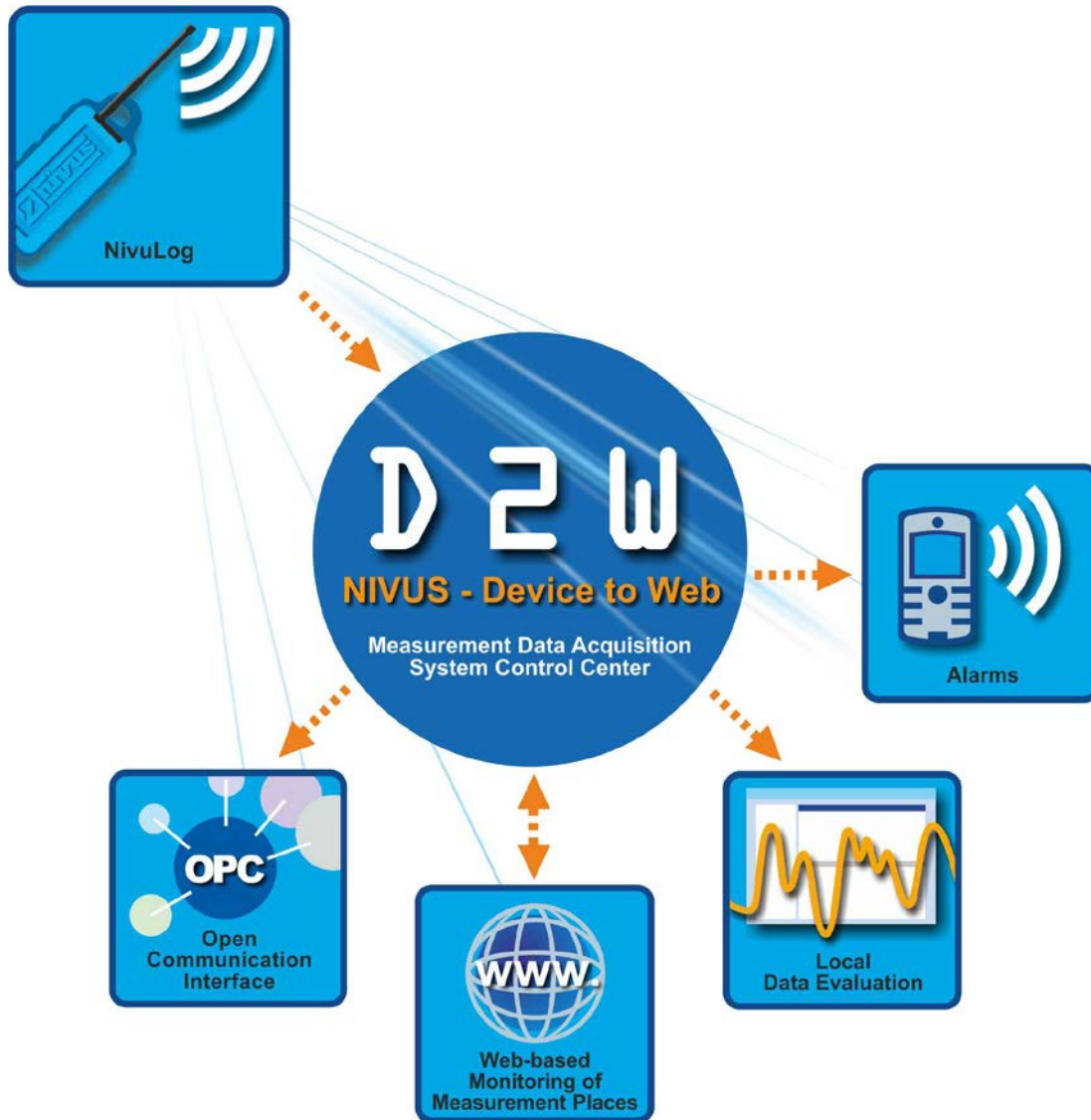


Technical Information On D2W - Device to Web Internet Portal

(Original – German)



H:\Produkte\I2W\I2W-xPath Interface-ti-00-en.docx

NIVUS GmbH

Im Taele 2
75031 Eppingen, Germany
Phone +49 (0)7262 9191-0
Fax +49 (0)7262 9191-999
E-Mail: info@nivus.com
Internet: www.nivus.com

NIVUS AG

Hauptstrasse 49
CH – 8750 Glarus
Tel. +41 (0)55 / 645 20 66
Fax +41 (0)55 / 645 20 14
E-mail: swiss@nivus.com
Internet: www.nivus.de

NIVUS Austria

Föhrenhaingasse 6
A – 2201 Gerasdorf bei Wien
Tel.: +43 (0)2246 32319
Fax: +43 (0)2246 32386
E-mail: austrias@nivus.com
Internet: www.nivus.de

NIVUS Sp. z o. o

Ul. Hutnicza 3 / B-18
PL – 81-212 Gdynia
Tel. +48 (0)58 / 760 20 15
Fax +48 (0)58 / 760 20 14
E-mail: poland@nivus.com
Internet: www.nivus.pl

NIVUS France

14, rue de la Paix
F – 67770 Sessenheim
Tel. +33 (0)388071696
Fax +33 (0)388071697
E-mail: france@nivus.com
Internet: www.nivus.fr

NIVUS U.K.

P.O. Box 342
Egerton, Bolton
Lancs. BL7 9WD, U.K.
Tel: +44 (0)1204 591559
Fax: +44 (0)1204 592686
E-mail: info@nivus.com
Internet: www.nivus.com

NIVUS U.K.

Wedgewood Rugby Road
Weston under Wetherley
Royal Leamington Spa
CV33 9BW, Warwickshire
Tel.: +44 (0)1926 632470
E-Mail: info@nivus.com
Internet: www.nivus.com


NIVUS Middle East (FZE)

Building Q 1-1 ap. 055
P.O. Box: 9217
Sharjah Airport International Free Zone
Tel.: +971 6 55 78 224
Fax: +971 6 55 78 225
E-Mail: Middle-East@nivus.com
Internet: www.nivus.com

NIVUS Korea Co. Ltd.

2nd Floor Taewha Plaza
570-1 Majeon-dong, Seo-gu
INCHEON 404-820
Tel.: +82 70 8620 7436
Fax: +82 70 8620 7437
E-Mail: korea@nivus.com
Internet: www.nivus.com

1. Basics

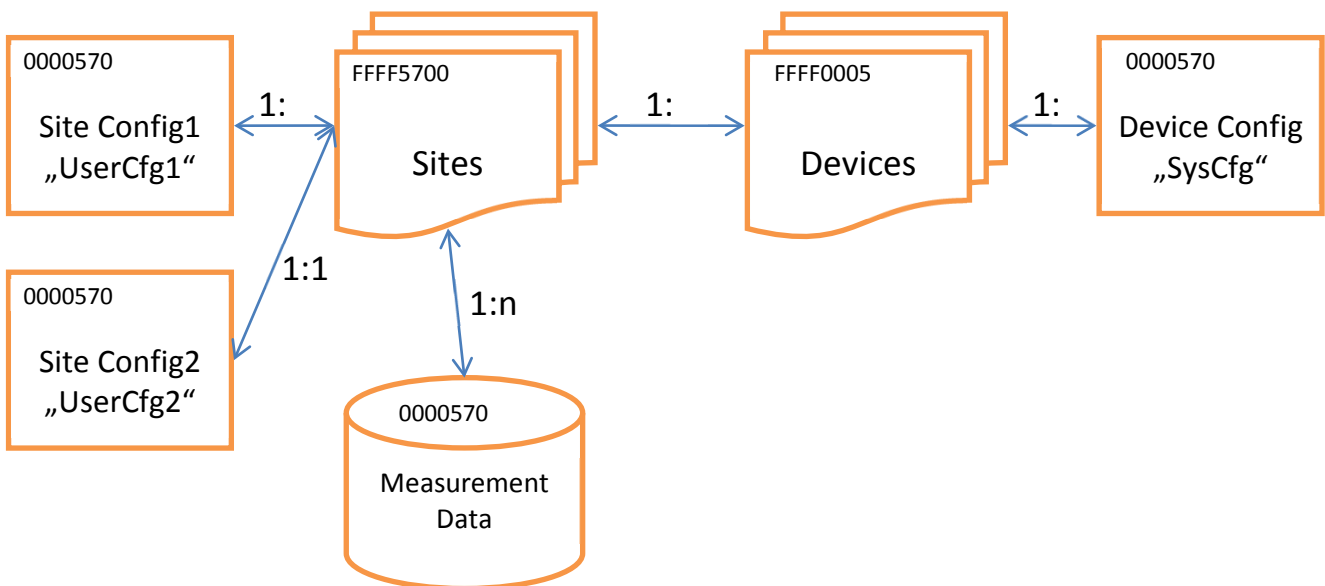
 The **xPath Interface** enables the export and import of data, settings and statistics in XML format. The data are determined using XPath queries.


1.1 Table of Content

1. Basics	3
1.1 Table of Content.....	3
1.2 Table Names	3
2. XML Export	4
2.1 Fields	5
2.2 Filters.....	5
2.3 Index Terms	6
2.4 XML Format.....	7
2.5 Site Types (<i>instrument classes</i>)	7
3. XML Import	7
3.1 Structure of an XML File	8

1.2 Table Names

The database is structured as follows:



 The table names in *hex Format* and are built up as follows (*x=device class*)...

Code	Description
0000xx00	measurement data
0000xx07	log
0000xx24	position (GPS) and network info
0000xx30	GPS – measurement data
FFFFxx00	site settings
FFFF0001	customers
FFFF0005	instruments
FFFF0010	connection list
“others”	instrument settings (depending on the device class)

Instrument	Class (x)	00	07	08	09	0A	0B	0C	0D	12	15	24	30
Instrumentspec.		✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
Sitespec.		✓			✓	✓	✓						✓
NivuLog Easy	52	✓	✓	✓	✓	✓						✓	✓
NivuLogPCM	3C	✓	✓	✓	✓				✓	✓		✓	✓
NivuLog2Ex	5A	✓	✓	✓	✓				✓	✓		✓	✓
NivuLog2Ex Legacy	2B	✓	✓	✓	✓				✓	✓		✓	✓
NivuLog4	27	✓	✓	✓	✓				✓	✓		✓	✓
NivuLogPLC	55	✓											
ImportCSV	51	✓											

2. XML Export



A *GET* command can be dropped e.g. with a common internet browser. The command line (*URL*) is built up as follows...

```
http://{IP-Adresse}:8080/{XPath Query}?{login}
```



The *wildcards* between the brackets are...

```
{IP-Adresse} ..... address of the server which runs the XPathSrv module (=D2W server)
{xPath Query} ..... XPath of the desired XML element (eg 00002600[stamp>='1.1.01']/ (temp))
{login} ..... a D2W server user account in order to authenticate the query (eg user=ul&password=p1)
```

The result is an *XML document*.



A listing of all fields of an element can be queried by a leading *@* before the table name ...

```
http://127.0.0.1:8080/@00005200?user=ul&password=p1
```

Result (all fields of the data element of NivuLog Easy instruments)...

```
<?xml version="1.0" encoding="windows-1252" ?>
<Result>
<Field title="Site">context</Field>
<Field title="Flags">flags</Field>
<Field title="Channel 1">ch1</Field>
<Field title="Channel 2">ch2</Field>
<Field title="Channel 3">ch3</Field>
<Field title="Channel 4">ch4</Field>
<Field title="Out 1">output1</Field>
<Field title="Out 2">output2</Field>
<Field title="Temp">one_wire_temp</Field>
<Field title="calc60">calc60</Field>
<Field title="calc61">calc61</Field>
<Field title="calc62">calc62</Field>
<Field title="calc63">calc63</Field>
<Field title="calc64">calc64</Field>
<Field title="GSM St?rke" units="dBm">gsm_level</Field>
<Field title="Battery" units="V">voltage</Field>
<Field title="Batterie [%]" units="%">battery</Field>
<Field title="Voltage" units="V">voltage_ext</Field>
<Field title="Temp" units="°C">temp</Field>
</Result>
```

2.1 Fields



If there is more than one field queried, they have to be put in round brackets () and separated by | or *blank*. If no field is specified, all fields are being output according to the user grants...

```
http://127.0.0.1:8080/00002709?user=ul&password=p1
```

Result (all User 1 configurations of NivuLog2Ex/4 units on the server)...

```
<Result>
  <User-configuration id="999A91AA7AD35F87" stamp="28.8.2007 13:39:22">
    <context title="Site">998B6C3C9C5F4B03</context>
    <measure_interval title="Measurement cycle" units="Min">1,00</measure_interval>
    <transfer_interval title="Transmission cycle" units="Min">10,00</transfer_interval>
    <mode1 title="Mode">0</mode1>
    <mode2 title="Mode">0</mode2>
    <mode3 title="Mode">0</mode3>
    <mode4 title="Mode">0</mode4>
    <high_level1 title="High Level" units="mV">0</high_level1>
    <high_level2 title="High Level" units="mV">0</high_level2>
    <high_level3 title="High Level" units="mV">0</high_level3>
    <high_level4 title="High Level" units="mV">0</high_level4>
    <tageswechselzeit title="Day change over">0</tageswechselzeit>
    <pos_interval title="Position cycle" units="hours">0,8</pos_interval>
    <warmup title="Warmup time" units="[sec]">10</warmup>
  </User-configuration>
  {...}
</Result>
```

Example (all PIN settings of NivuLog2Ex/4 units on the server)...

```
http://127.0.0.1:8080/0000270D/sim_pin?user=ul&password=p1
```

Example (all transfer and sample periods of NivuLog Easy units on the server)...

```
http://127.0.0.1:8080/00005209/(measure_interval|transfer_interval)?user=ul&password=p1
```

Example (label, street and ZIP of all customers on the server)...

```
http://127.0.0.1:8080/FFFF0001/(name|strasse|plzort)?user=ul&password=p1
```

2.2 Filters



A filter consists of terms which are combined by the logical operators *and* and *or*. The terms can be prioritised by round brackets (). In the topmost level only *and* joins are allowed. A filter consists of a field name, an operator and a constant. All fields and tables are allowed.

Attention: labels (eg *site names*) are case sensitive.

Hint: The filter `ref` must not stand between single quotes '.

Additional attributes available...

```
id ..... unique record identifier
stamp ..... time stamp of the record (only operators >= and <= allowed)
ref ..... unique record identifier of the reference record
```

Allowed mathematical operators... =, <>, >, <, >=, <=

Allowed data types to be used as constants...

```
Integer ..... figures
Float ..... floating point figures
Strings ..... text in single quotes '
Date ..... timestamp in single quotes '
```

Formats (decimal separator, date format, ...) are applied according to the user settings.



Example (list of all NivuLog Easy site names of a customer)...

Variant a... `http://127.0.0.1:8080/FFFF0001[name='Demo Kunde 1']/FFFF5200/(name){login}`
 Variant b... `http://127.0.0.1:8080/FFFF0001[id=98F2999AFF3FA5DD]/FFFF5200/(name){login}`
 Variant c... `http://127.0.0.1:8080/FFFF5200[ref=98F2999AFF3FA5DD]/name){login}`

Example...

```
http://127.0.0.1:8080/FFFF0001[name='Testschacht']/FFFF5200[name='Messgeraet
2']/00005200[stamp>='01.12.2008' and gas>5]/(){login}
```

Result...

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<Result>
  <myDatasensData id="FFFF01590001C311" stamp="22.12.2008 15:10:00">
    <battery>66</battery>
    <cputemp title="CPU Temp" units="°C">8,0</cputemp>
    <context title="Messstelle">9B2C60F35705322D</context>
    <gas title="Gas" units="ppm">9,40</gas>
    <gasraw title="Raw" units="mV">1.025,8</gasraw>
    <raw_calib title="Gas-Raw" units="ppm">9,42</raw_calib>
    <calc60 title="Rechenkanal 1" units="%"></calc60>
    <calc61 title="Rechenkanal 2" units="%"></calc61>
    <calc62 title="Rechenkanal 3" units="%"></calc62>
    <calc63 title="Rechenkanal 4" units="%"></calc63>
    <calc64 title="Rechenkanal 5" units="%"></calc64>
    <volt title="Batterie" units="%">66</volt>
    <gsm title="GSM Stärke" units="dBm">-95</gsm>
    <temp title="Int. Temp" units="°C">7</temp>
  </myDatasensData>
  {...}
</Result>
```

2.3 Index Terms



Allowed indices...

[0] *first record (with the oldest timestamp)*
 [last()-1] *latest record (with the latest timestamp)*



Example (latest record)...

```
http://127.0.0.1:8080/00002600[ref=98DEF2093233DC58][last()-1]/(gas){login}
```

Result...

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<Result>
  <myDatasensData id="FFFF01590001ECFF" stamp="30.12.2008 03:04:00">
    <battery>65</battery>
    <cputemp title="CPU Temp" units="°C">8,6</cputemp>
    <context title="Messstelle">9B2C60F35705322D</context>
    <gas title="Gas" units="ppm">0,00</gas>
    <gasraw title="Raw" units="mV">1.022,1</gasraw>
    <raw_calib title="Gas-Raw" units="ppm">0,00</raw_calib>
    <calc60 title="Rechenkanal 1" units="%"></calc60>
    <calc61 title="Rechenkanal 2" units="%"></calc61>
    <calc62 title="Rechenkanal 3" units="%"></calc62>
    <calc63 title="Rechenkanal 4" units="%"></calc63>
    <calc64 title="Rechenkanal 5" units="%"></calc64>
    <volt title="Batterie" units="%">65</volt>
    <gsm title="GSM Stärke" units="dBm">0L</gsm>
    <temp title="Int. Temp" units="°C">9</temp>
  </myDatasensData>
</Result>
```

2.4 XML Format



The result is an *XML document*, whose root is `<Result>`. Each record has at least one child element, and each child consists of the queried fields.

The child elements have the following attributes...

`id`.....*unique identifier*
`stamp`.....*timestamp of the record*

The fields have the following attributes...

`title`.....*title of the measurement channel (at measurement points depending on the according settings)*
`unit`.....*unit of the measurement channel (at measurement points depending on the according settings)*



Example...

```
<?xml version="1.0"?>
<Result>
  <Customers id="98F2999AFF3FA5DD" stamp="2007-03-20 10:49:44">
    <name title="Name">Demo Kunde 1</name>
    <strasse title="Street">Demo Strasse</strasse>
    <plzort title="Zip & City"></plzort>
  </Customers>
  <Customers id="98DEF16864DC6DAC" stamp="2007-03-21 14:39:42">
    <name title="Name">Demo Kunde 2</name>
    <strasse title="Street">Demo Strasse</strasse>
    <plzort title="Zip & City">6758 Musterort</plzort>
  </Customers>
</Result>
```

2.5 Site Types (*instrument classes*)



Example (*licensed instrument classes*)...

`http://127.0.0.1:8080/SiteTypes/()?(login)`

Result...

```
<?xml version="1.0"?>
<Result>
  <SiteType class="2700" title="NivuLog 4">FFFF2700</SiteType>
  <SiteType class="5200" title="NivuLog Easy+">FFFF5200</SiteType>
  <SiteType class="3C00" title="NivuLog PCM">FFFF3C00</SiteType>
</Result>
```

3. XML Import



- Only existing records and settings can be modified, and new records can be added. Historic data cannot be inserted.
- Make sure that the `stamp` of historic records is ascending. Otherwise problems may arise!
- When settings are *modified*, be sure to also update `stamp`, otherwise the settings are not synchronized between instrument and server.
- When records shall be *created*, `id=""` and `<tag>0000xx00</tag>` (=device class) must be entered.
- When records shall be deleted, `<tag>0</tag>` must be entered. Deleting historic data (e.g. measurement data) is not possible!

3.1 Structure of an XML File

Example (data modification/import)...



```
<?xml version="1.0"?>
<Result>
  <00002A00 stamp="11.03.2009 09:00" id="FFFF01590001ECFF"> (MODIFICATION)
    <rh>33,5</rh>
  </00002A00>
  <00002A00 stamp="11.03.2009 09:30" id="0"> (IMPORT)
    <rh>33,7</rh>
    <tag>00002A00</tag>
    <context>9ACDB2573445DE2B</context>
  </00002A00>
</Result>
```



The `id` attribute is a mandatory field.

When entering `id="0"` a new record is created with the desired properties. In all other cases the record with the desired `id` is being modified.

The `tag` element is a mandatory field when a record shall be created. It determinates the target table which will contain the data.

The `context` element determinates the referencing site record.

The property attributes (eg *title, units, ...*) are irrelevant for the import and may be omitted.

Hint: Formats (decimal separator, date format, ...) are being applied according to the user settings.