SOLVIMUS
METERING SOLUTIONS

FUNCTIONAL EXTENSION BY SCRIPTING

24.04.2020

The scope of functions covered by our data concentrators and gateways keeps rising continuously. Often a customer requirement would be a major driver for this. But, in this case, you always have to weigh up what to integrate into the standard, and what to implement on a specific basis and only for a given application.

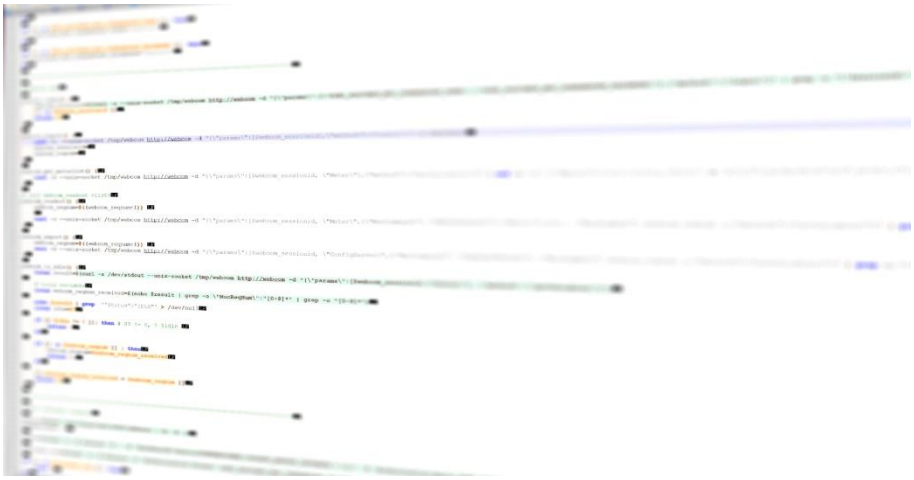
As every function integrated must still remain controllable and operable. We do not want to bother our customers by presenting them with an overloaded interface. To remain as flexible as possible in spite of this, some functions have only been possible by parameterizing them manually on a system level until now.

A complete changeover of the device families to the Linux operating system is now giving new opportunities to us.

For Linux, a very large number of tools, small programs, and libraries are available on an open-source basis. This is paving the way for developing an even more extensive or even more customer-specific product.

Of course, it is possible to find a software solution for many applications; they can be deployed and added to our devices for direct use. But this would be overly individual, and would require a specific adaptation for every purpose. It is not generic.

Contrary to this method, our generic approach involves using so-called scripts, i.e., small program snippets (macros) which allow to read, process, and output data while having only a limited scope of functions. The advantage is that no compiling will be required as the scripts are executed via an interpreter. This also enables users to adapt or create scripts by themselves.



As interpreters, we also use compiled standard environments like BASH or XSLTPROC. Scripts can run in these environments and perform various functions.

1 BASH Script

BASH is a command line interpreter. It makes the same opportunities available, which are also offered in the command line (Linux shell). Using variables, control flow elements (e.g. if queries), and arithmetic operations are possible in addition.

This allows to implement a very large variety of things, including e.g.:

- Sending out pings
- Setting the system time
- Sending data to an interface
- Retrieving information from a website
- Sending mails
- Restarting the system

But this list is far from complete. A complete listing of the commands available on our Linux system is very long, and also includes all-purpose tools like curl or openVPN.

A small example would look as follows:

```
#!/bin/bash
# parameter 1: IP of DUT
failcnt=0
succcnt=0
while true; do
# check device
 if ! ping -c 1 -w 1 $1 > /dev/null; then
# not present
   failcnt=$((failcnt + 1))
   wait=15
 else
# present
   succcnt=$((succcnt + 1))
 fi
 echo „$(date) device ping result: $failcnt vs. $succcnt“
 sleep $wait
done
```

It is started using an IP address as a parameter, and will then send out a ping to this IP every 15 seconds in an endless loop. Depending on whether this is successful or not, a counter will count up and output on a cyclic basis.

This should be enough to give you a first impression. More specific examples will be explained from time to time in upcoming blog posts.

2 XSLT Script

XSLTPROC is an XSLT parser which is integrated into our devices but is not as extensive and dedicated to the formatting of data.

XSLT is a script language used for transforming XML data to another structured format (XML or also CSV).

It basically allows adapting the output format of data to our data collector specifically for every customer. Internally we will process the data as XML, and thus allow to provide a basis for a script. The XSLT script builds upon this database, and then generates the corresponding output format required.

For more information on the opportunities XSLT offers, please refer to the following two blog posts: Data with format and stature – export of MUC data – part I and part II.

Regarding this topic, we will also explain specific examples in upcoming blog posts from time to time.