



What's New At Robelle



Jan 2018



Sharing Data HP-UX (BE) to Linux (SE)
Converting Free Format Numbers to Binary Data
Gavin Scott and the Current State of AI
Silicon Valley historians saddened over loss of Hewlett Packard archive in fire
Robelle Book Club

Sharing Data HP-UX (BE) to Linux (SE)

As you may know Suprtool / Open has been ported to Linux machines and it has been running for some years now. A fundamental difference between HP-UX and Linux is the Endianness or byte-order of binary data. For a quick summary of what this is take a look at the Wikipedia page on Endianness:

<https://en.wikipedia.org/wiki/Endianness>

In practical terms the main problem that this causes is that normally you would have to convert all data to ASCII in order to be able to read the data accurately. Some customers

have had a need to natively have files from HP-UX with binary data be moved over to Linux. Specifically without having to export to ascii and re-import into a self-describing file and have to redefine all the data etc.

Suprtool/Open now has the ability to read self-describing files and all of its binary data that describes the file (in Big Endian Format) and also the data itself (Integer and Logical Fields), and convert the data on the fly from Big Endian to Little Endian internally.

Suprtool/Open also can create .sd files and data files in Big Endian format so Suprtool and some versions of Cobol compilers on Linux that can read Big Endian Integers etc. This means that data can be moved over faster and natively.

The only conversion that needs to occur is with respect to the Self-Describing file, which we provide in the form of a small utility program call sdlinux. This utility will help convert an HP-UX self-describing file which will have Big Endian numbers, in the .sd file and in the integers and logicals in the data file.

Sdlinux, will change the sd version field which tells Suprtool/Open the endianness of the sd file.

SD Version	Endianness	Extended Names
B.00.00	BIG (HP-UX) LITTLE(LINUX)	NO
B.00.01	BIG (HP-UX) LITTLE(LINUX)	YES
B.00.02	BIG	NO
B.00.03	BIG	YES

The sdlinux utility has four options, -f, -r, -h and -d. Only one option can be specified at any given time:

Option	SD Version	Integer Field
-f	B.00.00 becomes B.00.02 and B.00.01 becomes B.00.03	Any integer/logical field gets flagged as BE.
-r	B.00.02 becomes B.00.00 and B.00.03 becomes B.00.01	Any integer/ logical field gets the Endianness flag turned off.
-h	B.00.02 becomes B.00.00 and B.00.03 becomes B.00.01	No effect.
-d	No effect.	Any integer/logical field gets the Endianness flag turned off.

The sdlinux utility gets run with the above options and a filename for an argument as in:

```
./sdlinux '-filename'  
./sdlinux '-rfilename'  
./sdlinux '-hfilename'  
./sdlinux '-dfilename'
```

The filename specified just needs to be the data file name not the sd filename. Regardless, sdlinux will figure out what to do if the .sd extension is in the filename argument.

Please see the following example how you can move data from HP-UX, to Linux natively without having to export to ascii. On HP-UX, you simply create a self-describing file as you normally would:

```
Base mydb  
Get mydataset  
Out outfile,link  
Xeq
```

Then you can use sdlinux to make some quick changes to the Self-describing information, specifically the version in the header and the integers and logicals get updated with the big endian flag being turned on:

```
sdlinux 'foutfile'
```

You can then ftp the data file and the sd file over to your Linux box:

```
ftp linuxbox.robelle.com  
user  
password  
put outfile outfile  
put outfile.sd outfile.sd  
quit
```

You can then reverse the effects of the `-f` option with the `-r` option:

```
sdlinux -routfile
```

Then on Linux you can read the sd file natively even though the sd information has big endian information and the data file can be read with bigendian integers and logicals.

```

>in outfile
>form
  File: outfile      (SD Version B.00.02)  Has linefeeds
  Entry:              Offset
  CHAR-FIELD          X5      1
  INT-FIELD           I1      6           BE
  DBL-FIELD           I2      8           BE
  PACKED-FIELD        P12     12
  PACKED*-FIELD       P12     18
  QUAD-FIELD          I4      24           BE
  ID-FIELD            I1      32           BE
  LOGICAL-FIELD       K1      34           BE
  DBLLOG-FIELD        K2      36           BE
  ZONED-FIELD         Z5      40
  FILLER              X36     45
Entry Length: 80  Blocking: 1
>num 3
>list
>xreq
>IN outfile (0) >OUT $NULL (0)
CHAR-FIELD          = 11111          INT-FIELD           = 1111
DBL-FIELD           = 11111          PACKED-FIELD        = +11111
PACKED*-FIELD       = +11111          QUAD-FIELD          = 11111
ID-FIELD            = 1              LOGICAL-FIELD       = 1111
DBLLOG-FIELD        = 11111          ZONED-FIELD         = 11111
FILLER              =

```

For more information on being able to natively share data from HP-UX (BE) to Linux (LE) let us know by E-mailing neil@robelle.com.

Converting Free Format Numbers to Binary Data

This past month support has encountered a number of questions about how to handle numbers being “imported”. We sometimes forget about all that the \$number function is capable of and I find it is helpful to review what we wrote in the manual to describe the capabilities

Suprtool has the ability to accept free-form "numbers" as display data types. This means numbers in the form:

```

1234.45-
-12345
-123.2134
12343
$123.45

```

can now be accepted and converted to any other numeric data type. Consider the following data:

Item-number	New-Price
12345	+123.45
34563	+ 27.5
21312	+ 1.545

Suprtool has the ability to read and convert the data in New-Price using the number function. Let's say we want New-Price to be a double integer and currently occupies eight bytes

starting in position six. Here is the task you would use to convert the New-Price free-format number into a double integer.

```
>in mynums
>def item-number,1,5,byte
>def new-price-ascii,6,8,display
>def new-price,1,4,double
>item new-price-ascii,dec,2
>item new-price,dec,2
>ext item-number
>ext new-price=$number(new-price-ascii)
>out somefile,link
>xeq
```

The \$number function take the free-format number and make it a valid display number. It will determine the decimal, sign and add leading zeroes. It will round the number to the defined number of decimal places.

In the case of 1.545 number, Suprtool will round the value to be 1.55, since the given number of decimal places is two and the preceding value is five or greater. If you have a whole number such as 54, with no decimal point the value becomes 54.00.

Suprtool will not accept data that has:

```
More than one sign.
More than one decimal place.
Spaces in between numbers.
Signs that are in between numbers.
Characters that are not over punch characters.
Fields that when edited do not fit in the defined space for the display field.
```

You can control the character that defines the currency, thousand and decimal symbol for other currencies and formats using the following commands:

```
>set decimalsymbol "."
>set thousandsymbol ","
>set currencysymbol "$"
```

Suprtool in the above case will strip the currency and thousand symbols and use the decimal symbol to determine the number of decimal places. You can set these characters to any values you want but the defaults for each are used in the above set commands. The Decimal and thousand symbols are only single characters. The currency symbol allows for four characters.

Gavin Scott and the Current State of AI

Recently I received an e-mail from longtime friend of Robelle, Gavin Scott on how he spent his winter vacation, which was researching the current state of AI. It is a great summary and introduction to AI on many fronts. The article begins below:

I set out to learn more about the current state of AI and specifically Deep Neural Networks over the past couple weeks, and I am here to report on some of the cool stuff I've found out.

A couple years ago Nvidia introduced their latest (at that time) GPU architecture called Pascal, and most of the announcement was spent talking about Neural Networks, which at the time seemed like a somewhat niche market compared to say all the 3D gaming applications, VR, etc. But now a new generation of neural-net driven “AI” is taking the world by storm, and in 2018 if you aren’t looking for opportunities to utilize these technologies in your application solutions then you’re doing it wrong.

In the past we have asked ourselves, “What are we going to do with all those MIPS in the future?” and now we have a new answer to that question, we’re going to expend them training up neural networks to solve hard problems for us. This is an area of technology where you need new heights of processing power to get good results, and the more teraflops you can throw at a problem the more ***interesting*** the solutions you can achieve. Things don’t just get faster, they get smarter, and some really amazing behavior starts to emerge.

So, what’s changed? AI has been a thing through a cycle of boom and bust (unrealized expectations) for several decades now, and the ideas behind a software neural network go back to 1943. First, it’s the availability of hardware that can grind through the obscene amounts of linear algebra required, and second it’s the software architectures for advanced neural networks that has been invented over the last five years or so. You’ve seen the fruits of this development in 2017, starting with Google Deep Mind’s Alpha Go that beat world Go champion Lee Sewol in March, something that pretty much shocked the world as almost nobody expected this to be possible for another decade or two.

Deep Mind has since followed up with AlphaGoZero which learned the game of Go from first principles by playing itself (the original AlphaGo had been trained using a lot of human game data), and then Alpha Zero, a general-purpose engine that became the most powerful chess player on earth after only four hours of self-play starting from just the basic rules of chess. You can read some of the chess world’s reaction to the games played by Alpha Zero that have led to an active conspiracy theory that Google must actually have secretly developed god-level sentient AI. To prove the flexibility of the Alpha Zero architecture, they fed it the rules of Shogi which it then mastered to a similar level.

In the past couple months, Google has had further breakthroughs in speech synthesis, and their latest paper suggests that computer generated speech which is indistinguishable from human speech is now pretty much a solved problem. Similar successes are being made in image recognition, content synthesis, and many other areas. Expect to start seeing things that amaze you, and machines to start being scary-smart at specific tasks. We’re in no danger of producing sentient machines in the near future, but a machine that can play poker or even engage in general negotiations better than any human is already a thing. Also understanding and generating text is another area that’s rapidly advancing, so expect spam filtering to become a hard problem again.

Another area that’s getting a lot of interest is GANs, Generative Adversarial Networks, where you create two competing neural networks, one that say tries to forge paintings and one that tries to detect forgeries. It’s possible for these to learn from each other and so both get better at their tasks without requiring supervised learning, to the point that both get impressively good at their respective tasks.

And lots of other “AI things” that have never worked worth a darn before (computers composing/generating music for example) are starting to finally give interesting results.

Ok, so most of us here don't get to work on things like game-playing software or facial recognition, etc. so how is all this going to help me post transactions to my general ledger SQL tables more efficiently? Well, it probably isn't, but there are potential applications to be found anywhere you have a source for lots of good quality data. That stream of GL transactions, or your customer transaction history, or web interaction log files, etc. can all potentially be things you can focus a Machine Learning application at.

2017 was the year of democratization for neural network software development. In the past, you probably needed at least a Master's in math in order to follow the literature in Machine Learning, and even then, a lot of the theory wasn't practical to implement due to hardware costs, and if you did want to implement something you were on your own to write all the code to do it. Now there are several very good, well documented, and newbie friendly libraries for Machine Learning, such as Google's TensorFlow. There's a big push to make these technologies available to the unwashed masses of software developers out there, and you can implement some extremely powerful applications in just a few lines of code. There are also lots of books and tutorial videos, as well as MOOCs you can sign up for.

It's all interesting and fun stuff, and very accessible now. It only takes a few minutes to set up a local development environment with Anaconda and TensorFlow, and you can be classifying MNIST digit images (the Hello World of Machine Learning).

In addition to having all the cutting-edge software available for free, the hardware needed to accelerate the matrix multiplications that are at the core of neural networks is very accessible as well. My gaming laptop has an Nvidia 1060 GPU in it, and installing tensorflow-gpu was straightforward and gave me something like a factor of 20x speedup. The 1060 has about 4 teraflops of processing power. And you can easily access clusters of GPU hardware through cloud services from Google, Amazon, and Microsoft, and Google is now offering cloud access to their new TPU (Tensor Processing Units) custom processor chip hardware for neural network training and execution (the AlphaZero chess engine runs using just four TPUs).

It's a completely different paradigm for programming. Rather than solve the problem yourself and teach the computer how to execute your solution in a traditional program, you just set up an environment in which the computer can learn how to solve the problem on its own and you provide examples for the computer to learn from. It really feels pretty magical. Finally, the damn machines are actually doing something useful!

Anyhow, I've had a blast and I strongly recommend learning enough about this stuff so you can understand it (since you're going to be running into it everywhere) and if you're still doing any kind of solution development then this is a powerful new tool for which you should be on the lookout for applications. It's also still "cool" at this point, though you probably only have this year or so to distinguish your solution through its use of "AI" before absolutely everyone is doing it. There are already Venture Capital firms which exist just to help incubate people with AI ideas.

Machine Learning gets my vote as the most interesting, useful, and accessible area of computer advancement right now, over things like Quantum Computing and Block chain-based applications.

Resources:

I started with this book:

<https://www.amazon.com/Hands-Machine-Learning-Scikit-Learn-TensorFlow/dp/1491962291>

which starts with basic ML techniques before moving into deep learning with TensorFlow. You can also get started with the very excellent tutorials and documentation on the TF site:

https://www.tensorflow.org/get_started/

Assorted readings / viewings:

<https://blog.statsbot.co/deep-learning-achievements-4c563e034257>

<https://tryolabs.com/blog/2016/12/06/major-advancements-deep-learning-2016/> (Nice overview of GANs)

<http://karpathy.github.io/2015/05/21/rnn-effectiveness/> (this one is a classic on RNNs)

https://www.reddit.com/r/artificial/comments/7lwrep/media_synthesis_and_personalized_content_my/

<https://www.youtube.com/watch?v=u7kQ5INfUfg> (make scary cats!) <https://affinelayer.com/pixsrv/> (try it yourself)

<https://www.youtube.com/watch?v=9VC0c3pndbl> (Turn winter into summer!)

<https://www.youtube.com/watch?v=aircAruvnKk> (short introduction to neural networks and how they work)

And as a bonus, playing with this stuff will pull you into the world of Python via Jupyter notebooks and Data Science where you should be for other reasons too.

Silicon Valley historians saddened over loss of Hewlett Packard archive in fire *startribune.com*

More than 100 boxes containing letters and other documents from Hewlett-Packard founders William Hewlett and David Packard were incinerated when the Tubbs Fire tore through one building on the campus of Keysight Technologies headquarters in Santa Rosa. Corporate historians say the loss goes far beyond the estimated \$2 million value of the collection. That's because it contained thousands of pages of history documenting the firsthand thoughts and strategies of the two tech pioneers who formed the electronics company in Palo Alto.

"It's heartbreaking," said Karen Lewis, a former HP archivist who pored through each of those boxes in the 1980s, cataloging each document to help preserve that rich record for future researchers and historians. While it isn't a perfect solution, and a daunting project to start, make a digital archive of all your records, photos, etc. You'd be amazed at what you can fit on an inexpensive external hard drive of just a few terabytes, kept in a safe deposit box or at a family member's house.

<https://www.thestar.com.my/tech/tech-news/2017/11/01/silicon-valley-historians-saddened-over-loss-of-hewlett-packard-archive-in-fire/>

<https://spectrum.ieee.org/view-from-the-valley/tech-history/silicon-revolution/loss-of-hewlettpackard-archive-a-wakeup-call-for-computer-historians>

Robelle Book Club

We Robellians like to read. In our last newsletter, we listed a few of the books we have shared and enjoyed.

We have listed a few more of our recent “good reads” below.

[Ragged Company](#) by [Richard Wagamese](#)

[One Brother Shy](#) by [Terry Fallis](#) (light fun read)

[Smoke and Mirrors: Short Fiction and Illusions](#) by [Neil Gaiman](#) (short stories)

[Dog Songs](#) by [Mary Oliver](#) (for dog lovers)

We are always looking for book recommendations. Send your book recommendation tammy@robelle.com

To unsubscribe from our newsletter, send us an e-mail at wudunsub@robelle.com

May the New Year bring fresh bout of excellence in your life