

DMV003 DMetashell & Phoscript Stack Machine

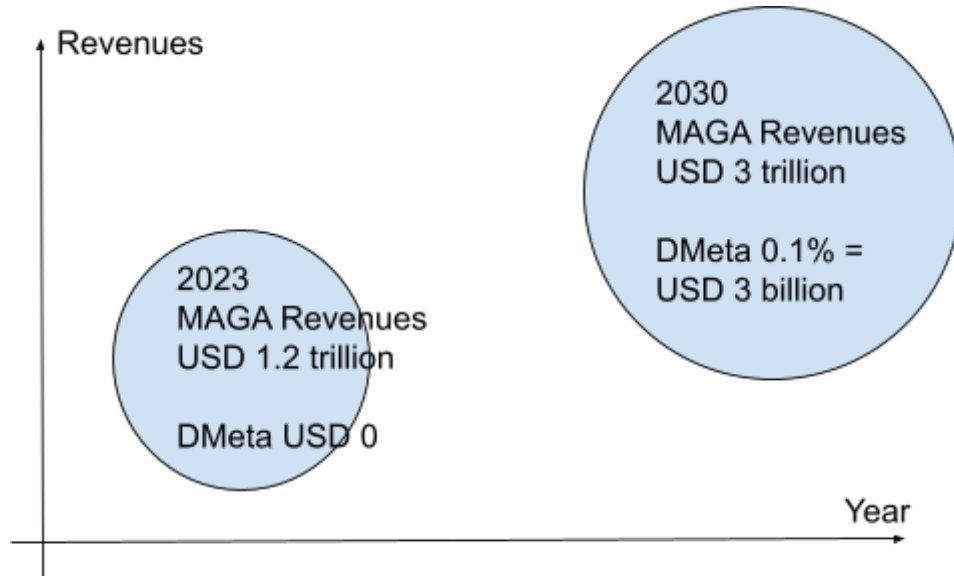


Figure 1: Projection of Revenues

In a previous article, we demonstrate how we may build a “Decentralised Metaverse” owned and operated by individual users and programmers, which may generate revenues of at least USD 3 billion by 2030, using a CONSERVATIVE estimate of 0.1% of MAGA revenues, where MAGA refers to Microsoft, Meta, Amazon, Google, Apple, although the acronym is a funny collision with the political one, as shown in figure 1.

- https://godmeta.github.io/doc/DMV002_Build_Decentralised_Metaverse_DMetashell.pdf

Figure 2 is a screenshot of a web browser's developer console. The console shows a series of log messages, including:

- 45: "Graph/hg/FTQSY996LA==:2023-11-06T17:49:52.438+0000 Command to hash then execute, macro xn"
- 46: "Graph/hg/HjY7NdMkyw==:2023-11-06T18:43:02.475+0000 2472 php phos.php comments Graph/hg/xhash fgc: dup: {\'\' swap: pos: substr: jd: 1 pick: i: sslh: 3 pick: je: s:'"
- 47: "Graph/hg/HvW0z08Tmg==:2023-11-06T18:44:01.041+0000 Pick 0 element of \$S, je: for easy editing in o_cdw.json"
- 48: "Graph/hg/AlV20bbKog==:2023-11-08T00:48:01.099+0000 2549 php phos.php hgnc: dup: /php.phos/ g: dup: ak: array: ON ECHO bv: : f1 dup: 5 pick: swap: 1 + i: array: swap: ap: swap: 5 pick: swap: i: ap: \\\; map: f1 je: enl:""
- 49: "Graph/hg/GsL0o1qtEw==:2023-11-08T00:48:22.594+0000 comments no color je:""
- 50: "Graph/hg/Dbm5UuwSmg==:2023-11-14T12:54:12.834+0000 command_not_found_handle() { echo "It's my handle!"; echo "Arguments: \$@"; php phos.php \$@" }
- 51: "Graph/hg/Et3ygPECUQ==:2023-11-14T12:54:32.324+0000 f(': cell a b input: th tn: : : row cell cell cell 3 ms: tr tn: : body getn: 0 i: div ce: row row row 3 ms: table tn: ih: ac:)'"
- 52: "Graph/hg/CoHymmq50Q==:2023-11-14T12:54:56.943+0000 3 x 3 rows of input cells in html"
- 53: "Graph/hg/GFprREH4TQ==:2023-11-14T17:44:03.933+0000 f(': B_F '+ JSON.stringify(M.S[0].\$CDW.l_1) +' jd: l_1 acdw: htab33 hgje ON ECHO bv: enl: ; AJE:)'"
- 54: "Graph/hg/C67mHyGG7g==:2023-11-14T17:44:59.852+0000 Back end sends JSON to front end, AJE push JSON to stack without display."
- 55: "Graph/hg/E-n7ky7BxA==:2023-11-15T11:55:05.960+0000 f(': B_F comments je xh ; AJE') // comments je has enl: at end"
- 56: "Graph/hg/BcZzR5x2cg==:2023-11-15T11:56:37.234+0000 Show comments to front end with AJE = AJAX Eval"
- 57: "Graph/hg/AlCtIM_CRw==:2023-10-23T15:35:48.198+0000 abc"
- 58: "Graph/hg/FTuLBbeXGg==:2023-10-24T09:49:05.782+0000 AlCtIM_CRw== def"
- 59: "Graph/hg/Fvhs4JFebQ==:{"t": "2023-11-05T22:03:12.070+0000", "msg_a": "Graph\\hg\\Hls0M3_Bvg==:2023-11-04T19:04:02.890+0000 What ever ever ever", "msg_b": "Graph\\hg\\FRdH6NB0xg==:2023-11-05T12:08:33.536+0000 I accept"}"
- 60: "Graph/hg/F9qY6zWq5Q==:{"t": "2023-11-05T22:48:42.719+0000", "msg_b": "Graph\\hg\\FRdH6NB0xg==:2023-11-05T12:08:33.536+0000 I accept", "msg_a": "Graph\\hg\\Hls0M3_Bvg==:2023-11-04T19:04:02.890+0000 What ever ever ever"}"
- 61: "Graph/hg/Dd3fV0l0zg==:{"t": "2023-11-06T01:22:04.735+0000", "msg_b": "Graph\\hg\\FRdH6NB0xg==:2023-11-05T12:08:33.536+0000 I accept", "msg_a": "Graph\\hg\\Hls0M3_Bvg==:2023-11-04T19:04:02.890+0000 What ever ever ever"}"
- 62: "Graph/hg/BiG_A5B1_w==:{"t": "2023-11-06T01:22:19.538+0000", "msg_b": "Graph\\hg\\FRdH6NB0xg==:2023-11-05T12:08:33.536+0000 I accept", "msg_a": "Graph\\hg\\Hls0M3_Bvg==:2023-11-04T19:04:02.890+0000 What ever ever ever"}"

The console also shows a prototype for an Array and a function call: `>> f(': B_F hgje echo ; AJE')`

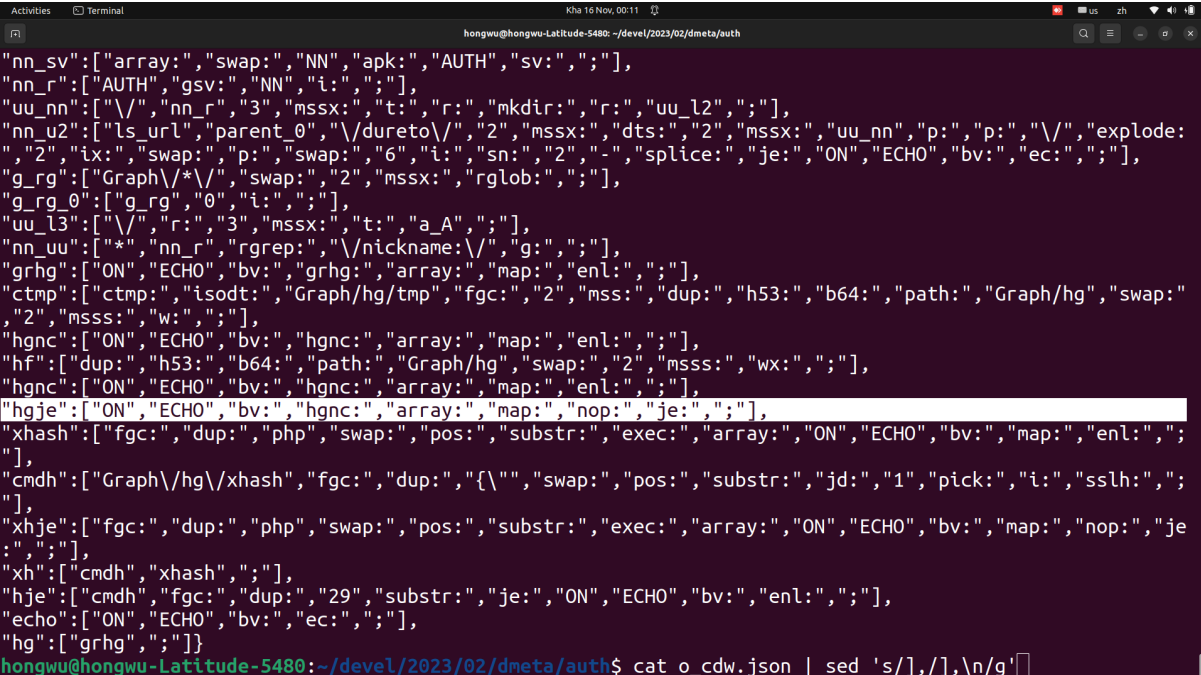
Figure 2

Figure 2 shows an output screenshot of running DMetashell, where the command was:

```
f(': B_F hgje echo ; AJE')
```

f() is a JavaScript function to initiate DMetashell, which in turn executes commands contained in the string. DMetashell runs Phoscript, a metaprogramming script derived from the FORTH programming language.

```
: B_F hgje echo ; \ backslash+space marks start of comment
\: .... ; colon and semicolon enclose a "word definition"
\ word in FORTH means function name
\ B_F is the function name
\ "hgje echo" are the subroutines in function B_F
```



```
hongwu@hongwu-Latitude-5480: ~/devel/2023/02/dmeta/auth
"nn_sv":["array:", "swap:", "NN", "apk:", "AUTH", "sv:", ";"],
"nn_r":["AUTH", "gsv:", "NN", "i:", ";"],
"uu_nn":["\\", "nn_r", "3", "mssx:", "t:", "r:", "mkdir:", "r:", "uu_l2", ";"],
"nn_u2":["ls_url", "parent_0", "\\dureto\\", "2", "mssx:", "dts:", "2", "mssx:", "uu_nn", "p:", "p:", "\\", "explode:",
"2", "ix:", "swap:", "p:", "swap:", "6", "i:", "sn:", "2", "-", "splice:", "je:", "ON", "ECHO", "bv:", "ec:", ";"],
"grhg":["Graph\\*/", "swap:", "2", "mssx:", "rglob:", ";"],
"grg_0":["grg", "0", "i:", ";"],
"uu_l3":["\\", "r:", "3", "mssx:", "t:", "a_A", ";"],
"nn_uu":["*", "nn_r", "rgrep:", "\\nickname:\\", "g:", ";"],
"grhg":["ON", "ECHO", "bv:", "grhg:", "array:", "map:", "enl:", ";"],
"ctmp":["ctmp:", "isodt:", "Graph/hg/tmp", "fgc:", "2", "mss:", "dup:", "h53:", "b64:", "path:", "Graph/hg", "swap:",
"2", "mss:", "w:", ";"],
"hgnc":["ON", "ECHO", "bv:", "hgnc:", "array:", "map:", "enl:", ";"],
"hf":["dup:", "h53:", "b64:", "path:", "Graph/hg", "swap:", "2", "mss:", "wx:", ";"],
"hgnc":["ON", "ECHO", "bv:", "hgnc:", "array:", "map:", "enl:", ";"],
"hgje":["ON", "ECHO", "bv:", "hgnc:", "array:", "map:", "nop:", "je:", ";"],
"xhash":["fgc:", "dup:", "php", "swap:", "pos:", "substr:", "exec:", "array:", "ON", "ECHO", "bv:", "map:", "enl:", ";"],
"cmdh":["Graph/hg/xhash", "fgc:", "dup:", "{\\", "swap:", "pos:", "substr:", "jd:", "1", "pick:", "i:", "sslh:", ";"],
"xhje":["fgc:", "dup:", "php", "swap:", "pos:", "substr:", "exec:", "array:", "ON", "ECHO", "bv:", "map:", "nop:", "je:", ";"],
"xh":["cmdh", "xhash", ";"],
"hje":["cmdh", "fgc:", "dup:", "29", "substr:", "je:", "ON", "ECHO", "bv:", "enl:", ";"],
"echo":["ON", "ECHO", "bv:", "ec:", ";"],
"hg":["grhg", ";"]
hongwu@hongwu-Latitude-5480: ~/devel/2023/02/dmeta/auth$ cat o_cdw.json | sed 's/./,\\n/g'
```

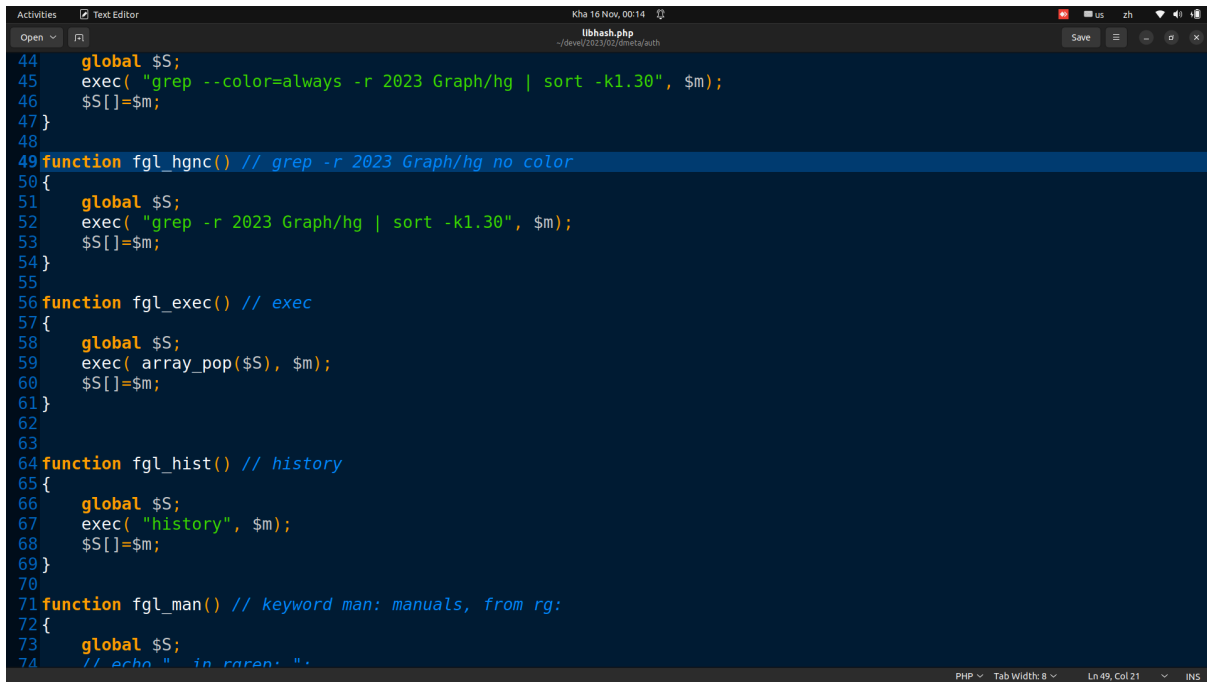
Figure 3

The definitions of hgje and echo are given in o_cdw.json, where cdw means “colon definition words”, as shown in figure 3:

```
"hgje":["ON", "ECHO", "bv:", "hgnc:", "array:", "map:", "nop:", "je:", ";"],
"echo":["ON", "ECHO", "bv:", "ec:", ";"],
```

o_cdw.json simply stores FORTH style colon definitions in JSON format.

Of these, the crucial functions are hgnc: which is mapped to fgl_hgnc() as shown in figure 4.

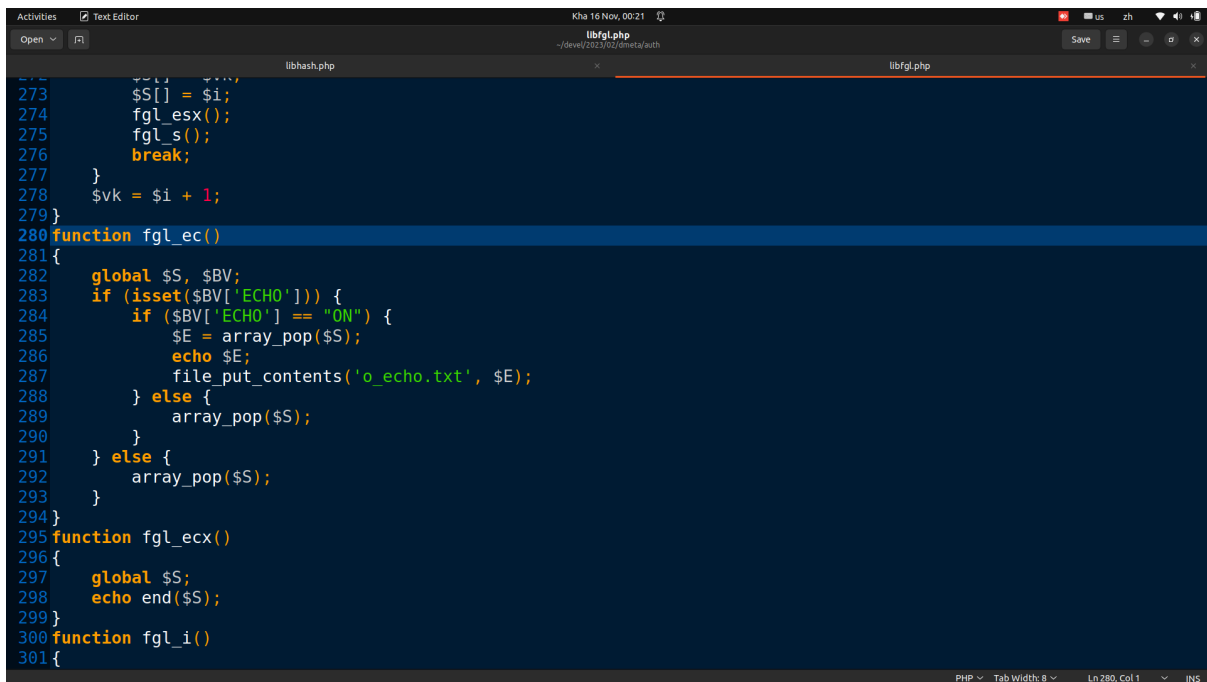
A screenshot of a text editor window titled 'libhash.php' with a dark theme. The code shows a function `fgl_hgnc()` at line 49, which is a recursive function that uses `exec()` to run a `grep` command. Other functions like `fgl_exec()` and `fgl_hist()` are also visible. The status bar at the bottom indicates 'PHP', 'Tab Width: 8', 'Ln 49, Col 21', and 'INS' mode.

```
44 global $S;
45 exec( "grep --color=always -r 2023 Graph/hg | sort -k1.30", $m);
46 $S[]=$m;
47 }
48
49 function fgl_hgnc() // grep -r 2023 Graph/hg no color
50 {
51     global $S;
52     exec( "grep -r 2023 Graph/hg | sort -k1.30", $m);
53     $S[]=$m;
54 }
55
56 function fgl_exec() // exec
57 {
58     global $S;
59     exec( array_pop($S), $m);
60     $S[]=$m;
61 }
62
63
64 function fgl_hist() // history
65 {
66     global $S;
67     exec( "history", $m);
68     $S[]=$m;
69 }
70
71 function fgl_man() // keyword man: manuals, from rg:
72 {
73     global $S;
74     // echo " in rg: ";
```

Figure 4

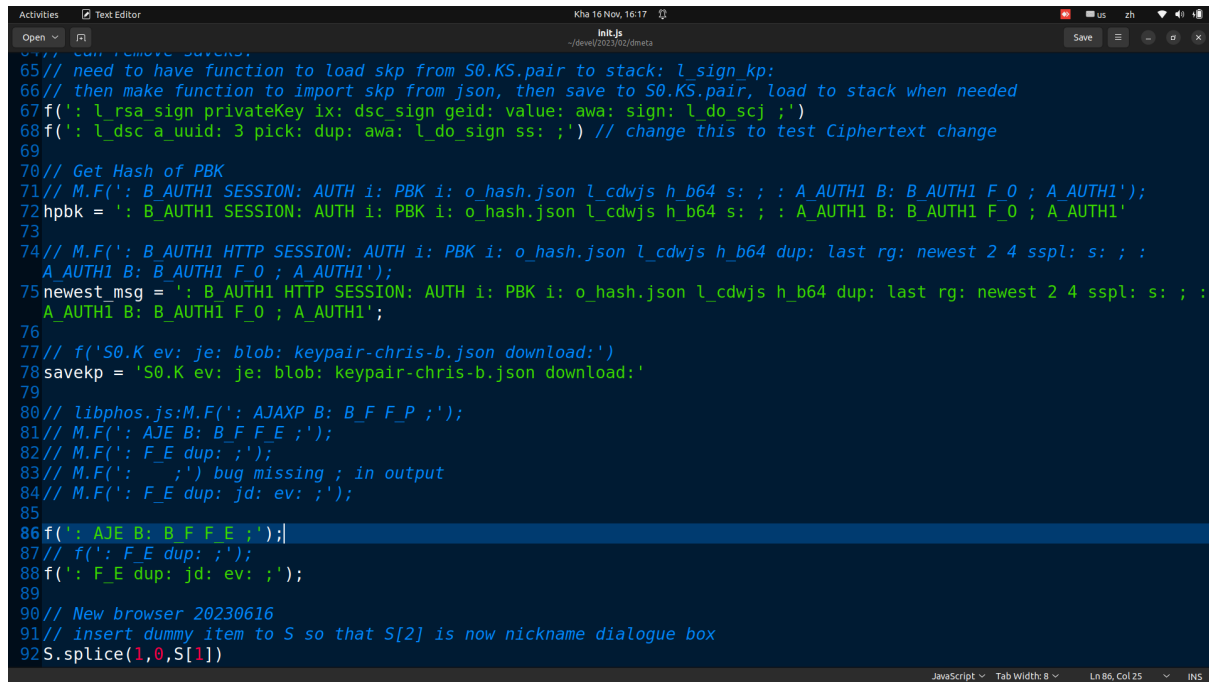
`fgl_hgnc()` is a PHP function which in turn executes "`grep -r 2023 Graph/hg | sort -k1.30`" which recursively displays all lines containing keyword 2023 in the subdirectory Graph/hg, then piped to a sort function. The output is shown in figure 2.

`echo` in turn calls `ec`: which is mapped to `fgl_ec()` in figure 5, which outputs the “top of stack” (TOS) item by calling PHP function `echo` in line 286.

A screenshot of a text editor window showing two files: 'libhash.php' and 'libfgl.php'. The 'libfgl.php' file is active, showing a function `fgl_ec()` at line 280. This function checks for an 'ECHO' environment variable and either echoes the top of the stack or pushes it. Other functions like `fgl_ecx()` and `fgl_i()` are also visible. The status bar at the bottom indicates 'PHP', 'Tab Width: 8', 'Ln 280, Col 1', and 'INS' mode.

```
273 $S[] = $i;
274 fgl_esx();
275 fgl_s();
276 break;
277 }
278 $vk = $i + 1;
279 }
280 function fgl_ec()
281 {
282     global $S, $BV;
283     if (isset($BV['ECHO'])) {
284         if ($BV['ECHO'] == "ON") {
285             $E = array_pop($S);
286             echo $E;
287             file_put_contents('o_echo.txt', $E);
288         } else {
289             array_pop($S);
290         }
291     } else {
292         array_pop($S);
293     }
294 }
295 function fgl_ecx()
296 {
297     global $S;
298     echo end($S);
299 }
300 function fgl_i()
301 {
```

Figure 5



```
65// need to have function to load skp from S0.KS.pair to stack: l_sign_kp:
66// then make function to import skp from json, then save to S0.KS.pair, load to stack when needed
67f('': l_rsa_sign privateKey ix: dsc_sign geid: value: awa: sign: l_do_scj ;')
68f('': l_dsc a_uuid: 3 pick: dup: awa: l_do_sign ss: ;') // change this to test Ciphertext change
69
70// Get Hash of PBK
71// M.F('': B_AUTH1 SESSION: AUTH i: PBK i: o_hash.json l_cdwjs h_b64 s: ; : A_AUTH1 B: B_AUTH1 F_0 ; A_AUTH1');
72hpbk = '': B_AUTH1 SESSION: AUTH i: PBK i: o_hash.json l_cdwjs h_b64 s: ; : A_AUTH1 B: B_AUTH1 F_0 ; A_AUTH1'
73
74// M.F('': B_AUTH1 HTTP SESSION: AUTH i: PBK i: o_hash.json l_cdwjs h_b64 dup: last rg: newest 2 4 sspl: s: ; :
  A_AUTH1 B: B_AUTH1 F_0 ; A_AUTH1');
75newest_msg = '': B_AUTH1 HTTP SESSION: AUTH i: PBK i: o_hash.json l_cdwjs h_b64 dup: last rg: newest 2 4 sspl: s: ; :
  A_AUTH1 B: B_AUTH1 F_0 ; A_AUTH1';
76
77// f('S0.K ev: je: blob: keypair-chris-b.json download:');
78savekp = 'S0.K ev: je: blob: keypair-chris-b.json download:'
79
80// libphos.js:M.F('': AJAXP B: B_F_F_P ;');
81// M.F('': AJE B: B_F_F_E ;');
82// M.F('': F_E dup: ;');
83// M.F('': ;') bug missing ; in output
84// M.F('': F_E dup: jd: ev: ;');
85
86f('': AJE B: B_F_F_E ;');
87// f('': F_E dup: ;');
88f('': F_E dup: jd: ev: ;');
89
90// New browser 20230616
91// insert dummy item to S so that S[2] is now nickname dialogue box
92S.splice(1,0,S[1])
```

Figure 6

The definitions of AJE are given in lines 86 and 88 of figure 6. Readers may refer to the code for detailed explanations. We shall simply give a summary here, where AJE is simply an AJAX function that stores the output from the backend (echo) on the top of stack in the front end, which was displayed in figure 2.

From the discussions above, readers may conclude that DMetashell consists of Phoscript stack machines running in BOTH the front end and back end. This is a breakthrough in web programming as we have now unified both front end and back end with ONE metaprogramming script, simplifying the learning curves for junior as well as senior programmers (who may have to learn multiple programming languages).

Further, Phoscript and DMetashell can be implemented in the equivalent of around 20 lines of JavaScript in ANY known programming language. So it can be ported to mobile apps, mainframe legacy systems or any other old and new systems easily. As such, it is actually a universal solution for many software development problems.