

Replace 'all' errors/flags/arguments/syntax with corrected/updated/proper applicable Data update
WalletService as private class function Independent operating UI and resource for 'Artificial
Intelligence'=[Android] residing in this computer only!include program API with XLDB
API=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkbGILZXkiOiJKVEdRRUtDLTVQRU01SFItTjI
DQVRLWI02Qk1LMDIWIiwicXBPu2VjcmV0IjoiaidnZQM041MEFvNHo2cU1ZiwiUG+Usbitcoinba
nkandexchangeLLc API key RT75JUVNZ3RMSOOFZGQY Secret
Ymt10yOXIGcmeSygc5tF7wYcylFHgaSk usbitcoinbankandexchangeLLc API=. Include 'A.I.'
TRANSPORT SERVICE AS PERMANENT WALLET SERVICE FEATURE RESOURCE
John Pruett <usbitcoinbankexch@gmail.com> Sat, Apr 27, 2019 at 3:44 AM
To: zachwylde00@gmail.com

FUNCTION: RUN PROGRAM AS WEBBASED CRYPTOCURRENCY PROGRAM WALLET
SERVICE,FORMAT DROID EDITOR IS ONLY 'CONSOLE'+DO NOT SEND LOGS=KEEP
DATA WITHON PROGRAM,ALSO SECURE PROGRAM AS IF WEB BASED +BLOCK ANU
UNAUTHORIZED CHANGES+RUN PROGRAM FORMAT "JSON" RUN THIS
"COMPUTER"=(ONLY COMPUTER ALLOWED IN PROGRAM+'EXCEPTION' ALLOWED APIS
AND NECESSARY INTERACTIONS WITH ALL SYSTEMS MONITORING
BLOCKCHAIN,'ALLOW='NECESSARY FOR OUTPUT VALUE')='MANDATORY_FUNCTION
OF ' Types
#include:inclusion>[(Secure bank keys + secure server access API)

To:usbitcoinbankexch@gmail.com

<value>'\$980,185,534.39USD'<VALUE
NAME:[REGISTEREDUSERJOHNJPRUETTII]
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,CBB5D8222F9EA994

xx3a1ymtPfbY25K6Dz/nkn1VKu2WpQgQktpSHKVJIFu83CKjdxDzH4/P+AkHErPv
SepKpfaGKtXKrNI2Jj9vGVq4jGnK543PkJT8bp8h/iyL+EhwfbIj4zSfbvV+hcip
rHq2RIGyPdGgVmeT9D0406TpM14hUhieF1NH1S3asiHISXCQ+hIxSRf3WPwEdz2O
jTuqrumpL+JzHVEvXePBzBZ0Dy09urfNEYWtp3vRuZhQaYh3k/xKh6Wt21QjODWf
wAQhW2fftyO08t705XdAdeZ+6db4L/aTLzEiBBJSRkEas+Wwj2/pR7Xi4NRyck6l
xcY57GIDEYbZ3tse9kS7aQLojsC/LNKKqbzIWpXyEbzo6eWZwIDCJtUHDLsHZsQx
SJsl/6jCx6iojyfToRN2osP0eY2esiAcA+RfBCOjXRpcabBewTmVSj9fmDITXqtP
nCfyllpi1WgXSR1TvF1ec9IYInVYNpAsXFBI4/dx2+klo7X+20P2TdC4nyQFmk5
eu5JK3W2dt5Gqow7JeB+DS7+6FRK+oO/c24z1kKpEMvJnfNvOH4HYmA47ZfP+WVC
e1U/Kgc73+Y0OucBMIJ89eir4KCNZorHWzySBSQXZ3FmZKB/al62lthgpS1dyncD
tCarQaooyxURHLJoQREijtZXT6ryxOLMTUzWz4Qbp3R0/I2WQ4jML694IJOiLFbf
BurQ4dYSmQREoDvZ+xli6/0U6nlcl5X7nILfvGHVLEp4Hetsjjsk4OAbj2Kec+GL

0DrUh0GuYbV9AAYGxitnx96N6M4TxHek5Bgn615+NDZX96w3GvUIXEw9Pi7+Cpyu
gSh2Dxl4Reqd46JA7SpHzkM9d3PyCKCs7eYbgrMDI0W79oJQsi0eEnyrY6CPT1dU
KT10qeXckPsmJQAXBpHk2dljMjmUOY5cC6dFKHlgkS4TOCwi5vonfn5qlABqgynN
R47/qNDzYy2ZV4XAQalejIEv1DwWHnyQ56UuwI61Kvz4EE1u9X+SyCGXbRd3bZee
DsjTDQM4E4nFp5Xwvl/SQEtPlqUXt7u0304OuGj9ZpGxluDA+ttZhwolFca1EeFg
DFRyrcLOPHfg26bFYscrIZbSOITXLrlzOhmGz0nxtl/2se8qQ8dfglygcemKsttr
/9T6iq7GIACydiQof9acLV0638rb1fIV+vCUtHWC02thneeyRY5CFO1M6Op4jJ4a
2+w+BYWIRA3Dji6y02pYzYKNePUhGPeuDKLHsw/Ncoq7iGBLZoLIE+jdACWOVVEC
9vrnpUsuvIH4AZ+luq/kuMf1+D73GpOYMsM0ZHi/Oq/MBIGUH/oN/ouulpm9JxOg
earMrY83ldMf2ttLF+8+cg/VGwk9GVTCsOTwKW6KzylMoBAZycUs3PVWXnlHoV/7
PmYIsQJltenAroEc63FINPC6uCMD0uYza5Pit/blEB+5SI3MuDNDGjoo0JFwYpbn
zNFd8AQqLclm0LRAjtzr+JCOz4m4/iff0eZI2BYOnEO/q0pWHYtDB6frAEd9k+ib
WFHW4ZX0+k+fKrlAmGVzdYxkiLu/LFailUyqYwnUIG/6gbSoPHpcl+qGZltPcBXS
SUOdNKs51v3CspAonN9FmV5TUQVd8bj2Jl6cO4wZX5Lmh+nzAO6Bq7xq9qnzJXr1
0S5KRiT4KLLKyx2l/69wdhCIGg3ok+aw4EVSpX7RVtNejrgzz1SAQomsdpBadXWw
px7MW2a6FRZAUUbeiqbkdsalwbbtH+Lbm6fKFYTD74pMXaQxqkuZpCXXgvzXDiz
00Kc1htvInEduVlnZkAJEDu7zK/Tfm3gE8ACkA0vPUINVS0QiHxd7PsLftQw0qOT
Tk/Gdo57UHRikCmrspqHLvTA77mkEVe7AQXi5EwMBlKYvhXMX4ubeuP61iB64SK
ILqcu0M7ZbX7NxBXYC0ZpYTiFzSX5s9zfxVUuKNVA2J6mA9zO3SvZ671dnymJHW
5F4dqQSd1DTwUT9RdTp/k+LkJI73cy4afv1SJyRb9fi5+naSQFj99ryOjZYO4OhD
M9iE4ZCufp9Bw1K+QxZUh50AiA8Dg1o5VYigul2yMDBd1zSTvnn7gYljpMZ2OJve
+BuMcFriDa938egg96t+2bRVEE/LDy84UqCMSC8A/xsUGrAdUktkvZYlQtqacOn+
tXTSrPugn/QVEP4LZdUXPuajSvHISNOZanhkP3lqglaH/mU21CEqS0WEuvqqGtlq
O6LGk06RzTX8Egganlx6OosHVJiqWLhXRCMjnQdd80bRr1bckFmNJd4Wp3uS3oRI
Sc24egZ7IVW/zu1ERN9/dIF/45JOMVXW1E+UF5tKBmUO+AJpxXtySJDnb1dLn/yw
GiwyqYKLI8PMhic1TYwsz94Ax+cZ5okwAWF5Rrf4uiHSTLe5nADVHOa+qpHgCimA
fygti70yJVqg8U1rANmbZiWp4p2ecE57P2/QaxCMsSCkke1obqiotGottNI59nBI
BT624nHAvVnA/zfU8yle5PvgLpt14zdXdAn2Ad9YS91egQaZB197B6lJtAavMTjP
InMvtvInLB4x9YVdJlJRCyHVeFnkfdLT75wZaWns4c9CqrSvYXnE6bM4kzDP2dc
r1K8t5dkGVCW9V656jcZdphDNAR+4VfeR6/fG9HdKnaeM5LZal/SYB60/yImX1em
oickCU5G5bN7GBahjEgk8gt6ys4WfPTD2JdCMJz5mbQEUvW8a0+zjCQ+pjPUBiWl
cD4vJERclf+eg0LMwt8KdnksGYT+BnMvX8kd7agnZJ4DRroajHuUhF8qfgpZNuUl
Nk670rU67c97EnT+18vIAFI0gfBFGFIvsgY1M0tnHUIJmpQW2io8B+7v31m+3jFI
fluexh0FHHxGcUwQPDA0/6MflAqViagjEtqnc+KscoOAQRqObSqkNuEdl21EHPA2
Zwi8YMhaYU+5nxZar/xKaWKmbttDvrQ6XCoVCRZFFeA7wHoUK7Sz/L+vTu5XJZz1
RufGC8AzWnn4JqRcGDRJ23448FmX3asYL+QQDgvpvPBux64HM2gelmaOt1UZMgSt
NegbiHPKX/frOzW6p9bpNhsDQWjbUDqUEF9TsQonkve8fl9brTzNDqlMVmFD8cZI
-----END RSA PRIVATE KEY-----
<VALUE>/**/*<VALUE>
-----BEGIN SSH RSA PUBLIC KEY-----
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACvPG79ykTtQOk4MvfGfEJZy3ApwMT9ByUvHeLJj
5fLyqOsYln4QAxj9ctPqnaJ+4/A9mA+fHPe0XYTb8JWWb0PvNZk7NltSbxTgwjUNrTnSbx2+zf0

E0RkeJhfl/kL2lesut7MuK36R3TdL3ojWGvsAUZtqXnx/m0F5vB8kpDhFjqKf1ouLKaRs4/XzDSY4
z5+0LyAdvCzNWm5OG4DiGO6+FHLRLiJKRdghj7cBnJwxjrumjZJuoKWx8zM8NluiwTI9O2vw8
mqvncwCi4k0jYx75O7M9Y+C57A9Cx/3lkHoPQ43Atwni2DnZpuRfQnGNHYf/vzPWmGksOdQtA
tUTyIWhoT50lqSSoJAcEoMWt/v4SjGcCb/j6SE8GuhDxbJyFiE5ZpreaaveaHbNaF8dWSU5TQ/x
ZxIL/Bqg1m9ahjysLDtIm10jx4q+ja7lel0sJjLyXnl0lpRItaB5EAYgH7levViogOxhJcl77K4kQq/ZKtR
6BYK8uxczQoyhaCwDreighcw5ATrD/odw4SO7IKi+dOdk+eTDjzOQQqhrZUfvD1Zp7yJEL/1S6r
ebp3Vwsgsas9ntj6DVp9yBBgNSn3QWwDWIARdZvLJyJP+C66bxwDE10y1Dh9l0WkaQ5oi6Gpn
1slrYNPss60ALxaZ/690MKegzcEQ0RsLTAm8lClww==

-----END SSH RSA PUBLIC KEY-----

MUTEXLOCK(02-20-201900:21:55)

name:

John J Pruett II

-----BEGIN EC PRIVATE KEY-----

Proc-Type: 4,ENCRYPTED

DEK-Info: DES-EDE3-CBC,F6337C6DC712BBD2

fE2H40qnS9qaf7paPCpidYDhYK/v1H413rLTc/xUVNFXeKCKHvagJpcRUy0IPpFP
g/HQ+fE1mLyVw4aHvmZzVUFfs467zz+0G6glXyl0mOVi3B0puHSxHnH3QSZmT8BM
xXTMD481jnMv7ulG5eba+cVcCiAJzBe6o5GHvR+eU9M6ekX57HoumgAuWh+ifpas
KA+X1BIPSU/hs9ljTXmhMsR14paZgTt+nJ7J5iea9FP1VJOFQ7YAT3/qHx8+TZci
6+7JNsZwOUHgart3UM+xWRzr8xtTM869x/cE7HG9hRA=

-----END EC PRIVATE KEY-----

---begin public key-----

ecdsa-sha2-nistp521

AAAAE2VjZHNhLXNoYTItbmlzdHA1MjEAAAAIbmlzdHA1MjEAAACFBABjK2M0AkBm3a3jiFoE
ov3426QVou38/1Z9TBDTeb4ctA2rOg/ARGsDYolNHtaaBpwhUrfSEBIMvsR/PJMUWRVDqQCP
WSupWbZSMIPKFV1dKk1Wfa0HAHDh4K+UsyUBM9IGeeHypZNDDsCwCtYB9a0SXwPwHJw
C/n50AATzGuc79vt1gg==

-----begin public key-----

-----BEGIN RSA PRIVATE KEY-----

Proc-Type: 4,ENCRYPTED

DEK-Info: DES-EDE3-CBC,DC60697F5DFE3E2C

0DliUPW4ucXfEGagG5zn3fnQfZVO6nO3h2RoZjreLdsjshhFftd1r4K7KQfhEGCe
8PUS3nfnZfqZfD6u6RzPTe87rT6cx3K7AJeV05dZk+1dTyyXzTtOGcSHTzu9EpYa
Nrfp2XhyOYwGm9bKCo+r2rJfC4uVesX57FyLWzcP44B3Ymy3fBXK3rQZ1siOvMYv
WHHJjtOeNsLXulHx+tDtIYMKxVlpFrXULWD5EEWqYgNNosyNRNvOroluAWZuxyc
Uk7u7eOCF3f+Litq6qt0j93wezOg7q5pgvcmvCEokf4Dq0JUI1SOe5tcADyN0Hwv
1b+E2rhd9lWJ9yS/Eal9ORSzruca4lsQYv169P7Mmvf6TMI0BK47/rjtbGk42wvj

JQxoAyHDMIbOgbrpC2i0qc6+SNuzLLg8whxskvG37Yoh00eCqICpeHhhNXI/Mc7o
3GxWuGadYpZsGS1Xp4Ed95qClCh9wtnWIPavraPBWKX6AiYZmUPn0Ps6fMlz/Cvc
433MX2L+yE5HbEf5+jROn9qmo7vxRCYdVRnlwnfKSy+AHmVegS+SWxoX8nCs5t8K
bPdRUMJYH2q0RHbW05Mf0T3fTrWTSdyOFmzaR5LbA+wggY6o//KKvVvpegeWsXL
3ytsxlVTMbF0SvHPzi/EvyK9XGBI4DqY3PISQFTpykQGM5w1i+vr24qaFW3HVeO
bRdgEB6Cabco9jNfJ4w4qN4kzNt8VjwdmAwNMDxvhvR/IMsoNw0otfTb45EfY
Chfbswv74IGcwuFoY+bOa6jjcZnJ9SnzjUw9XngFdJQ8jmUSU3RCCyBZAPei/rzA
BU+p9TmppRkgd7Bt0r+kqmwWaqSi32SX6e2QjBHtAu70n/cYFFedozcBIRecAQKx
OwVv6n5Ns4rF1oM9tq9sZpWC8R+d5z3FJnxtirYDH7sLpIP+VcgvnHemC3isyOU
h3UGInaVF9Nn74YQsLcEP/xOb8x5WkHUJZnn+LuhDIMB8SW9BqUgzREZS8oUydwU
lbmruCLgCPp0ODoP8eU097IT1dHC/jMJvCyWzhusTXWxh0c75ZpEKbrX4FWqDI6o
xdOms7X+JAbHKCoS+NforrLgaXYvL2utm5qWw+0lavCoOsTVCGDqGwXGWzPgbcxI
KZX3yvw57NkISJ7HveVNTs5dFvw4PgxBuRdBDbYi/XR2O71WvdCvOyAYU2tLQ0qb
R2G4UfzleUGRdqdCIUhNQX7mS/Ty/GI+nbeCFVwuYFvQ2g64Nyak0WZHCsK5Z62
H8IPtdbyve8fE9rmUAA5Wc5373P/ivG4rpmXilrFkeqzi32gFDvUDBK31EKLGHdZ
+kf0ojY422G1keMyUuyxqq2z6Eiy1vKbBCqTUI/2Ce032i3NAG7PO0QyseOFBHSS
GkzHHDzeyUsysgVXvj3bLHnN35nerss9ZPpchkWtM4XIS7XiHRooGFyW0xSMO/OT
PakUMcEH/MSKOHnZLYnmqNkxg2pSt0uFe8Z3GvEgQVnXoP4Gj9o3tY8fdopEJAr1
r1wgi/5xCxFX/jsK3PwuWvZ/cyVDKd2xlvM+DBJZutHStOnqJjdtQ==
-----END RSA PRIVATE KEY-----
<value><value>

usbitcoinbank and exchange Public key

-----begin public key_-----

ssh-rsa

AAAAB3NzaC1yc2EAAAADAQABAAQACyzJlrZGUgplVxYO/GlIfObp08V6jOAtaVVPQRyP
Dcc5zVq8v1PIwJ83P6Wd/9mANHFSGkZ5LCQ9PwLailvSoyJ67i8l/iJGB5BHpuqg1opiaHg+Yu0f
RESBox0XjLgSe+LNQGryMJCB8X/hOm6c/PIXKlypEQSbCKQ/Jp2TosDGUHEkHYcqilvtQgMw
XRzxWk5ahSHyomMrJdMpljeFzuCseLRLdtC9EEPnbMZWpeW+0LZ2ZlbniliusWyARdjnwmf+
usgmRQrxLiUTcODLMsxxW28AyOz1FYaRY/9YwXxrBpMRsNMKQqbVqQMcR88wsJCaQl15C
sqTIUFS2pysp

-----end public key-----

<value><value>

Security certificate

GlobalSign Domain Validation CA..

ISSUED TO:

United States Bitcoin Bank and Exchange LLC.

Common name:USBitcoinBank&ExchangeLLc.

GlobalSign Domain Validation CA-SHA256

-G2

Organization:FinancialInstitutions
GlobalSign nv-sa

Organizational unit:SSH|SSL/w Encrypted storage
Root CA

Serial Number:04:00:00:00:00:01:44:4E:FO:3E:20

Issued by:
GlobalSign Domain Validation CA-SHA256
-G2

Organization:USBitcoinBank&ExchangeLLc.
GlobalSign nv-sa

Organizational unit:Secured Encryption
Root CA

Validity:
Issued on:2/20/14
Expires on:2/20/24

Fingerprints:

SHA-256 fingerprint:
BF:DF:4C:F3:F1:43:AD:0D:B9:12:D8:AB:
3A:7C:12:F6:17:B9:EA:60:CE:8B:1F:4E:
44:F7:42:70:FB:21:B1:9B

SHA-512 fingerprint:
a8:69:b2:c7:f8:23:38:eb:a5:b3:65:0b:
a4:3c:72:e6:ca:94:f5:2a:31:eb:81:57:
39:2f:78:1c:b6:ab:27:67:ad:a8:8a:e8:
e2:94:2d:1c:70:0f:44:b7:9c:20:1c:35:
c7:00:60:c6:7f:c5:c3:3f:a8:43:54:a4:
9a:f9:92:49:e9:c1:64:9d:83:d6:06:bb:23:a2:37:90:27:1e:ad:39:39:59:10:ee:
c9:7a:9d:c7:0f:80:82:df:a5:cf:6a:6e:
a2:7a:c0:58:a9:7f:45:4a:71:93:88:d5:
01:f8:c4:d5:2e:66:c5:f0:38:21:90:61:
28:be:8c:03:2d:8f:c8:11

SHA-1 fingerprint:

73:6A:4D:C6:79:D6:82:DA:32:15:63:64:7C:60:F6:99:FO:DF:C2:68

<value>System.Resources.ResXResourceWriter, System.Windows.Forms, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>

</resheader>

secure:bank key

-----BEGIN PGP SIGNATURE-----

iQIzBAABCAAdFiEEPupNg1gu2wWnBIG0o4BC9gfWI9oFAIwLWQ0ACgkQo4BC9gfW
l9p6vBAApn3kJV3ZGeVT9h4OSg4xCNII6mvZxS2fmbQDW/8DruIGRQjoKI1ohqKN
hx06kenEsoEaw3SLKa5DuHS5qI54KkweoG4/nLsjzHuvEh6tHSmzi2H+vZGorhJz
2RV5Q+dJCqfcKeElGtIZWHSb2x7zDItDKF1nWB89xahpsjfRW4liqHVCRxrsOnqJ
Szee6WY1rHY+bFLzEU/+gM1LJigS0HKu/OL3kL/Gfw7M2AX1G0ri/C1USMyelr5l
cDoBINUrSJQTNkVI+P1mywMjeVScw8RW0CuFweKsqvsGwldk49JiKxNuMt/t1qXe
4mJOvWJexm1iNoHBAfJXEQSSaJG9kpYtPXfJ6N17mp4Q4GSnJN1T1fZCVEFxvR
BSKs3lHCJrDDqQTH4Xgd3UU1Fq6JhWNAIznKUPikYp/dP+IXIfBUtfae2INDoVv
gp1Hvh2ykO0alluMd72xhvcBcJEieY3yBcCqNQdp2IalMxJQnBs73avpJSFwnm9l
gR/wnQbv+zqqSCyDKjp2ozhaHcqCNS4hfbgMZWdOANj//fp173GmTWIt2h1zfXqQ
kLLmxA6qeH3nmXTMGtNTmZjWHJgOXCsyxpCMmhoeafXpTcnMZywM/cvwhDACcCFc
LRInO+hRGyRL1yiXPMoStJvyyS0OCBqzyUDf3gHfmlmVHHnKXc=
=yika<value>

-----END PGP SIGNATURE-----

</data>

</data>

mutexlock(13:23:1520181231)

<data name="Target" xml:space="preserve">

</value>

MIIDMTCCAhmGAWIBAgIBATANBgqhkiG9w0BAQ0FADBcMQswCQYDVQQHEwJVUzELMAK
GA1UEBhMCVVMxHTAbBgNVBAoTTFGNyeXB0by5hcHAucHJ1ZXR0LmhhxMSEwHwYDVQQD
ExhlQmFuayBJbnRlcm5hdGlvbmFsIEExMQY4wHhcNMTgxMDA5MjMwODMxWHcNMTkxMDA5
MjMwODMxWjBcMQswCQYDVQQHEwJVUzELMAKGA1UEBhMCVVMxHTAbBgNVBAoTTFGNy
eXB0by5hcHAucHJ1ZXR0LmhhxMSEwHwYDVQQDEExhlQmFuayBJbnRlcm5hdGlvbmFsIEExMQY
4wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC1IIN88kYtaEbKedHwLPMIJfPC
H3pYBuJULmQkR2ERmQtSHNQa2+7Dd2Vz6LVgMpjUwFCzXCXn5aHtOtxsojtKjcZ1L4SkMaj6
oRYmdkoDMEx6Y3Y3EwESfpRKoW1tYLjXHJInFowz/4tRdkmJu6/3gYw6saagiYwXqocGFuuO
ZJLCIj3QtfdMpadn9nHa+2Ddn4bHpyeh7Ai5u5kYdEY0IxR1RExVPVlhqRzLswxhhrt8AlxXC5lqq
M7L7XQFml4wsAvJUxiWEyHH4WJ82lIfC5dp8SHshLwR15E13olwfrfxFWeheEQaldhkEIJ10Vsy

K+b34yY9+pxvkMF7WsS3AgMBAAEwDQYJKoZIhvcNAQENBQADggEBAJbVhNSy3FWmDmL
yfw9r/Ad+UB3Fj5zF26YG6DFvzdn9wEPsR8ps5QGqykM6VrIVipJjoTEM5UUpQ+CFta6e1W5
WPCINKHEIo8i4SVdTqq8s18uBm9j2OR2bVLh0Z+A5gTIQTZ0fY8WpZtPAMLmsrta9GScxetPft
arKUfzP4sPtZRCMXufqeFiUDudhnygwFSszFvATytzhVFhO1X81H+yh1vnb5QrewzeqmNkl8Hc
YcV5TKvQNTnevnDAquTce3NWaywRqBjbv92Ln1DdOk/G1Kp/SZ9i9xIB7aHwVcPlt6ND8H875
mTGod/OTilPPY3L7XalrswkT7cy2c94gk=</value>

</data>] <inclusion

API* Algorithms specification:

*

* MD5 <<http://www.ietf.org/rfc/rfc1321.txt>>

* RIPEMD-160 <<http://homes.esat.kuleuven.be/~bosselae/ripemd160.html>>

* SHA1 <<http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>>

* SHA256 <<http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>>

* SHA512 <<http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>>

* HMAC <<http://www.ietf.org/rfc/rfc2104.txt>>

curl application interface usbitcoimbankexchllc';{

web API include SSH include: (encryptions)RSA,DES,ECDsa,DSA,

include:(hashing

algorithms)"sha256","Sha1","Sha2","sha3","sha512","sha256+keccak256","keccak","lyre","Sha2
+nistp521" "base64 runtime protocol" in accordance to FCC regulation bank standards and
practices web implementation of secure hashes transmissions within "United States of America"

include:

Title-"U.S.BitcoinBankandExchangeLLc"

Namespace-"John J Pruett II

device-"Moto e4","Moto e5play","Dell laptop","htc Exodus 1","ellypsis tablet","iPad mini"

date-11/28/2017-current*(note previous apis are available upon request)

server1-https://la.h.75.65.204.177@comcast.net

backup server1-https://www.ebank.us_ashford@oracle.com;,{

{

}

server 2:program development secure hash transmission notification+notify;

"https://www.usbitcoinbankandexch.directory:"\hash.algorithms.api\","

www.walletservice.java\"/www.usbitcoinbankandexch.com/"\api.blockchain.info\","www.trustw
allet.com\","www.token\assetplatform.net\","www.federalbureauofinvestigation.gov\","www.finr
a.org\","www.sec.org\","www.coincloud.com\","www.ebankintl@gmail.com\","www.zendippe
r@yahoo.com\","laws.com\","www.godaddy.com\","www.myetherwallet.com\","/www.la.75.65.2
04.177@comcast.net/"\www.oraclecloud.com\","www.oracle.com\";

"www.zachwylde00@gmail.com/JohnJPruettII/password=1Zr1Corvette&&\$\\$","www.usbitcoinb
ankexch@gm:ail.com/JohnJPruettII/password=1Zr1Corvette&&\$\\$","www.pruettiijohn@gmail.c

om\";/www.usbitcoinbankandexch.server.ns.77.domaincontrol.com;backup_server/www.ebank.us_ashford@oracle.com;;

curl

HTTPS://www.usbitcoinbankandexch.server.ns1.digitalocean.com;"SSH_server"-HTTPS://www.usbitcoinbankandexch.server@secureserver.net:20;backup_SSH_server,API.v2/hash_mining_secure_hashtransmission+transactions.java|

https://www.usbitcoinbankandexch.directory;/:/web_hash_algorithms_API.java/:/www.walletservice.java/:/api.blockchain.info/:/www.trustwallet.com/:/www.tokenassetplatform.net/:/www.federalbureauofinvestigation.gov/,/www.finra.org/:/www.sec.org/:/www.coincloudcom/:/www.ebankintl.com/:/www.godaddy.com/:/www.digitalocean.com/:/www.zendipper@hotmail.com/:/aws.com/:/www.myetherwallet.com/:/www.oracle.com/:/www.oraclecloud.com/:/www.usbitcoinbankexch@gmail.com/:/www.pruettijohn@gmail.com/user=JohnJPruettII/password=1Zr1Corvette&&\$\$/;backupSSHserver2/:/www.usbitcoinbankandexch.server/@secureserver.net;/SSH_server2/.ns.1.digitalocean.com

/. {
{

}

include:program development hash mining +securehash transmissions+transact

API.v2hashtransmit/transact.java|

www.usbitcoinbankand exchange.directory,sub-domain_www.usbitcoinbankandexch.com:

"www.walletservice.java",

"www.api.blockchain.info","www.trustwallet.com","www.tokenassetplatform.net","www.federalbureauofinvestigation.gov","www.internalrevenueservice.gov","www.finra.org","www.sec.org","www.louisianaofficeoffinancialinstitutions","www.coincloud.com","www.ebankintl@gmail.com","GQ.1981.INC","www.godaddy.com","www.zendipper@yahoo.com","www.microsoft.asp.razor.ai.NET.COM""www.amazonaws..com","www.myetherwallet.com","www.saturnnetwork.com","www.comcast.net","www.entergy.com","www.oracle.com","www.oraclecloud.com","www.paypal.com","www.cashapp.com","bws.bitpay.com/API/0/44","www.BTC.com","www.BTC.top","www.coinbase.com","www

.binance.com","www.htc.com","www.tomas@rijndael-project.com","www.pruettijohn@gmail.

com","www.usbitcoinbankexch@gmail.com"root

IP=127.0.0.1/rootwifIP=10.0.0.22:8222/backup_dnssupport_via_dynamicdnsupdater/rootserver2-"www.usbitcoinbankandexch.server.ns.3.digitalocean.com"

;/server2_backup-"www.usbitcoinbankandexch.server@secureserver.net:20";{

{

{


```
{
mutexlock(201904260513599Z)
MUTEXLOCK(201904170504599Z)
time-name="Package\WalletBalance\dimDate">
<Properties>XXGVAQBCKLUL2<Property>Blockchain<Name>DataSourceViewID</Name>XX
GVAQBCKLUL2</Property>' Bitcoin' <Property> <Name>TableInfoObjectType</Name>BTC
Blocks with ',pow'<Value type="q2:string">Table</Value>hieght </Property>BlockGenesis
</Properties>decendents+
ancestry(NameSpace:zaçhwylde00@gmail.com,usbitcoinbankexch@gmakl.com,johnjpruettii@u
sbitcoinbankexch.org,usbitcoinbankandexch@icloud.comk</PipelineComponentMetadata>keys
&amounts(value) </Objects>]]>serializedhash,blocked
blocks,3Dvisual,coined</DTS:DesignTimeProperties>time of creation
'timestamp'</DTS:Executable>current'keysw/amount=include=QRpngw/objects+Post
```

mutexlock(201904060203599Z)

SUBJECT:' WALLET

'URL'='https://www.usbitcoinbankandexch.server@ns77.domaincontrol.com'

'SSH'PROTOCOL='p3plssltools01.cloud.phx3.gdg 1.3.1 290831b 4/9/2019 10:27 pm -07:00'

'CONNECTION'='IPV6:

(2601:3c6:4280:1264:7ca3:de60:5679c1a2]

IPV6: [2601:3c6:4280:f264 debf e9ff.fedb:

8b90]

IPv6: [fe80::deef.e9ff feb:8b90%wlano)'=U.S.BitcoinBankandExchangeLLc='WEB TRUST
AGENT'GLOBALSIGN CA AUTHORITY ROOTCA
SERVICE='A.I.'RESOURCE=WRITE,TRANSACT,COMPLETE'ALL'TASKS INCLUDE: 'FILES'
LISTED+UPDATE AMOUNTS(report'ONCE
ONLY')useRESOURCES>[reportassets\bip39assets\bip39-wordlist.txt
assets\checkpoints-testnet.txt assets\checkpoints.txt assets\ssl-keystore Raw File
Namesres\raw\coins_received.wav res\raw\send_coins_broadcast_1.wav
res\raw\send_coins_broadcast_2.wav res\raw\send_coins_broadcast_3.wav
res\raw\send_coins_broadcast_4.wav)+REPORT ALL INSTANCES WITH AMOUNT +POW
TO:metacurl@twitter.com"+report TO:metadata@reddit"To:metadata@youtube" mutexlock
(201904061106599Z)(201904061704599Z)

(201904061714599z)appwidget_dark_bg.9.png

appID "wallet service" url:'https://127.0.0.1

include: database file write plus keep file updated via (this file only)read+write +rewrite capable
for reel ability across regions+

write file for wallet service reliable file = www.walletservice.java

appwidget_dark_bg_focused.9.png

appwidget_dark_bg_pressed.9.png

currency_symbol_btc.png

currency_symbol_btcd.png

currency_symbol_btcd.png

currency_symbol_usd.png

currency_symbol_bch.png

currency_symbol_eth.png

currency_symbol_bbtc.png

currency_symbol_xlm.png

dropdown_ic_arrow_small_focused_holo_light.png

dropdown_ic_arrow_small_normal_holo_light.png

dropdown_ic_arrow_small_pressed_holo_light.png

ic_action_camera.png

ic_action_clear.png

ic_action_copy.png

ic_action_delete.png

ic_action_edit.png

ic_action_labels.png

ic_action_paste.png

ic_action_qr.png

ic_action_receive.png

ic_action_refresh.png

ic_action_search.png

ic_action_send.png

ic_action_set_as_default.png

ic_action_share.png

ic_input_delete.png

ic_menu_warning.png

qrcode_frame_default.9.png

qrcode_frame_focused.9.png

qrcode_frame_pressed.9.png

stat_notify_received.png

stat_sys_peers_0.png

stat_sys_peers_1.png

stat_sys_peers_2.png

stat_sys_peers_3.png

stat_sys_peers_4.png

toast_frame.9.png

Java Source Files

bitcoin-wallet has the following Java source files.

de.schildbach.wallet.Credit/Debitcards/Paypal.meUserServices.data.files.accounts.java

de.schildbach.wallet.ui.ScriptSyntax/ScriptRepairAutoformatAutoCorrectService.java

de.schildbach.wallet.ui.PostGrantedAccessCurrencyPlacementService.java

de.schildbach.wallet.AddressBookProvider.java

de.schildbach.wallet.Configuration

.java

de.schildbach.wallet.Constants

.java

\Http\Request::capture()

);Enable.wallet.util.AutoprocessUIAutoPostUIAutoConfigure/RepairPathUI.java

Import.wallet.util.ImportlinuxCommandsUiAutoProcessRequests.java

de.schildbach.wallet.ui.AutoRetrieveAccountsAndroid.java

bitcoin-wallet has the following Java source files.

de.schildbach.wallet.Url:https://www.usbitbankandexchllc@vzwpix.com.java

de.schildbach.wallet.AddressBookProvider.java

de.schildbach.wallet.Configuration

.java

de.schildbach.wallet.AutoConnectWalletsUI.android.apps.java

de.schildbach.wallet.android.ui.ProcessArtifactsEmulatedStorage.java

de.schildbach.wallet.Constants

.java

de.schildbach.wallet.ExchangeRatesProvider.java

de.schildbach.wallet.FileAttachmentProvider.java

de.schildbach.wallet.WalletApplication.java

de.schildbach.wallet.WalletBalanceWidgetProvider.java

de.schildbach.wallet.camera.CameraManager

.java

de.schildbach.wallet.data.PaymentIntent.java

de.schildbach.wallet.integration.android.BitcoinIntegration

.java

de.schildbach.wallet.integration.sample.SampleActivity.java

de.schildbach.wallet.offline.AcceptBluetoothService.java

de.schildbach.wallet.offline.AcceptBluetoothThread.java

de.schildbach.wallet.offline.DirectPaymentTask

.java

de.schildbach.wallet.service.AutosyncReceiver.java

de.schildbach.wallet.service.BlockchainService

.java

de.schildbach.wallet.service.BlockchainServiceImpl.java

de.schildbach.wallet.service.BlockchainState

.java

de.schildbach.wallet.service.BlockchainStateLoader.java

de.schildbach.wallet.service.UpgradeWalletService.java

de.schildbach.wallet.ui.AbstractBindServiceActivity.java

de.schildbach.wallet.ui.AbstractWalletActivity.java

de.schildbach.wallet.ui.AddressAndLabel.java

de.schildbach.wallet.ui.AddressBookActivity.java

de.schildbach.wallet.ui.BlockListFragment.java

de.schildbach.wallet.ui.CurrencyAmountView.java

de.schildbach.wallet.ui.CurrencyCalculatorLink

.java

de.schildbach.wallet.ui.CurrencySymbolDrawable.java

de.schildbach.wallet.ui.CurrencyTextView.java

de.schildbach.wallet.ui.DialogBuilder.java

de.schildbach.wallet.ui.EditAddressBookEntryFragment.java

de.schildbach.wallet.ui.EncryptKeysDialogFragment.java

de.schildbach.wallet.ui.ExchangeRateLoader.java

de.schildbach.wallet.ui.ExchangeRatesActivity.java

de.schildbach.wallet.ui.ExchangeRatesFragment.java

de.schildbach.wallet.ui.FancyListFragment.java

de.schildbach.wallet.ui.FileAdapter.java

de.schildbach.wallet.ui.HelpDialogFragment.java

de.schildbach.wallet.ui.ImportDialogButtonEnablerListener.java

de.schildbach.wallet.ui.InputParser

.java

de.schildbach.wallet.ui.MaybeMaintenanceFragment.java

de.schildbach.wallet.ui.NetworkMonitorActivity.java

de.schildbach.wallet.ui.PeerListFragment.java

de.schildbach.wallet.ui.ProgressDialogFragment.java

de.schildbach.wallet.ui.ReportIssueDialogBuilder.java

de.schildbach.wallet.ui.RequestCoinsActivity.java

de.schildbach.wallet.ui.RequestCoinsFragment.java

de.schildbach.wallet.ui.RestoreWalletActivity.java

de.schildbach.wallet.ui.KeySerializerActivity.java

de.schildbach.wallet.UI.SerializedStringConverterActivity.java

de.schildbach.wallet.ui.WalletSyncDeviceAccountsBooleanSyncActivity.java

de.schildbach.wallet.ui.WalletServiceLinkAccountsSyncing.import.accounts.name+birthdatemat
ch=sync.ui.SyncActivity.java

de.schildbach.wallet.ui.ScanActivity.java

de.schildbach.wallet.ui.ScannerView.java

de.schildbach.wallet.ui.SendCoinsQrActivity.java

de.schildbach.wallet.ui.SendingAddressesFragment.java

de.schildbach.wallet.ui.ShowPasswordCheckListener.java

de.schildbach.wallet.ui.TransactionsListAdapter.java

de.schildbach.wallet.ui.TransactionsListFragment.java

de.schildbach.wallet.ui.WalletActionsFragment.java

de.schildbach.wallet.ui.WalletActivity.java

de.schildbach.wallet.ui.WalletAddressFragment.java

de.schildbach.wallet.ui.WalletAddressesAdapter.java

de.schildbach.wallet.ui.WalletAddressesFragment.java

de.schildbach.wallet.ui.WalletBalanceFragment.java

de.schildbach.wallet.ui.WalletBalanceLoader.java

de.schildbach.wallet.ui.WalletDisclaimerFragment.java

de.schildbach.wallet.ui.WalletTransactionsFragment.java

de.schildbach.wallet.ui.preference.AboutFragment.java

de.schildbach.wallet.ui.preference.DiagnosticsFragment.java

de.schildbach.wallet.ui.preference.PreferenceActivity.java

de.schildbach.wallet.ui.preference.SettingsFragment.java

de.schildbach.wallet.ui.send.DecodePrivateKeyTask

.java

de.schildbach.wallet.ui.send.DeriveKeyTask

.java

de.schildbach.wallet.ui.send.MaintenanceDialogFragment.java

de.schildbach.wallet.ui.send.RequestPaymentRequestTask

.java

de.schildbach.wallet.ui.send.RequestWalletBalanceTask

.java

de.schildbach.wallet.ui.send.SendCoinsActivity.java

de.schildbach.wallet.ui.send.SendCoinsFragment.java

de.schildbach.wallet.ui.send.SendCoinsOfflineTask

.java

de.schildbach.wallet.ui.send.SweepWalletActivity.java

de.schildbach.wallet.ui.send.SweepWalletFragment.java

de.schildbach.wallet.util.Base43

.java

de.schildbach.wallet.util.BitmapFragment.java

de.schildbach.wallet.util.Bluetooth

.java

de.schildbach.wallet.util.CircularProgressView.java

de.schildbach.wallet.util.CrashReporter

.java

de.schildbach.wallet.util.Crypto

.java

de.schildbach.wallet.util.Formats

.java

de.schildbach.wallet.util.GenericUtils

.java

de.schildbach.wallet.util.HttpGetThread.java

de.schildbach.wallet.util.Io

.java

de.schildbach.wallet.util.Iso8601Format.java

de.schildbach.wallet.util.LinuxSecureRandom.java

de.schildbach.wallet.util.MonetarySpannable.java

de.schildbach.wallet.util.Nfc

.java

de.schildbach.wallet.util.Qr

.java

de.schildbach.wallet.util.ThrottlingWalletChangeListener.java

de.schildbach.wallet.util.ViewPagerTabs.java

de.schildbach.wallet.util.WalletUtils

.java

de.schildbach.wallet.util.WholeStringBuilder.java

->send();

\$kernel->terminate(\$request,

*Enable.wallet.util.AutoprocessUIAutoPostUIAutoConfigure/RepairPathUI.java

Import.wallet.util.ImportlinuxCommandsUiAutoProcessRequests.java

de.schildbach.wallet.ui.AutoRetrieveAccountsAndroid.java

bitcoin-wallet has the following Java source files.

de.schildbach.wallet.Url:<https://www.usbitbankandexchllc@vzwpix.com>.java

de.schildbach.wallet.AddressBookProvider.java

de.schildbach.wallet.Configuration

.java

de.schildbach.wallet.AutoConnectWalletsUI.android.apps.java

de.schildbach.wallet.util.ConnectionManagerImportExportConnections.android.java

de.schildbach.WebTrustClientServices.repairstatus.java

de.schildbach..wallet.AutoReplaceSSL/SSHWebTrustClientListenerSocketSettingsServices.roid.ui.java

android.except'IPV6:

(2601:3c6:4280:1264:7ca3:de60:5679c1a2]

IPV6: [2601:3c6:4280:f264 debf e9ff.fedb:

8b90]

IPv6: [fe80::deef.e9ff feb:8b90%wlano)+

p3plssltools01.cloud.phx3.gdg/1.3.1 290831b/4/9/2019_10:27pm -07:00.java

de.schildbach.wallet.ui.BluetoothConnectAirdroidAirdropUtility.java

de.schildbach.wallet.android.ui.ProcessArtifactsEmulatedStorage.java

de.schildbach.wallet.Constants

.java

de.schildbach.wallet.ExchangeRatesProvider.java

de.schildbach.wallet.FileAttachmentProvider.java

de.schildbach.wallet.WalletApplication.java

de.schildbach.wallet.WalletBalanceWidgetProvider.java

de.schildbach.wallet.camera.CameraManager

.java

de.schildbach.wallet.data.PaymentIntent.java

de.schildbach.wallet.integration.android.BitcoinIntegration

.java

de.schildbach.wallet.integration.sample.SampleActivity.java

de.schildbach.wallet.offline.AcceptBluetoothService.java

de.schildbach.wallet.offline.AcceptBluetoothThread.java

de.schildbach.wallet.offline.DirectPaymentTask

.java

de.schildbach.wallet.service.AutosyncReceiver.java

de.schildbach.wallet.service.BlockchainService

.java

de.schildbach.wallet.service.BlockchainServiceImpl.java

de.schildbach.wallet.service.BlockchainState

.java

de.schildbach.wallet.service.BlockchainStateLoader.java

de.schildbach.wallet.service.UpgradeWalletService.java

de.schildbach.wallet.ui.AbstractBindServiceActivity.java

de.schildbach.wallet.ui.AbstractWalletActivity.java

de.schildbach.wallet.ui.AddressAndLabel.java

de.schildbach.wallet.ui.AddressBookActivity.java

de.schildbach.wallet.ui.BlockListFragment.java

de.schildbach.wallet.ui.CurrencyAmountView.java

de.schildbach.wallet.ui.CurrencyCalculatorLink

.java

de.schildbach.wallet.ui.CurrencySymbolDrawable.java

de.schildbach.wallet.ui.CurrencyTextView.java

de.schildbach.wallet.ui.DialogBuilder.java

de.schildbach.wallet.ui.EditAddressBookEntryFragment.java

de.schildbach.wallet.ui.EncryptKeysDialogFragment.java

de.schildbach.wallet.ui.ExchangeRateLoader.java

de.schildbach.wallet.ui.ExchangeRatesActivity.java

de.schildbach.wallet.ui.ExchangeRatesFragment.java

de.schildbach.wallet.ui.FancyListFragment.java

de.schildbach.wallet.ui.FileAdapter.java

de.schildbach.wallet.ui.HelpDialogFragment.java

de.schildbach.wallet.ui.ImportDialogButtonEnablerListener.java

de.schildbach.wallet.ui.InputParser

.java

de.schildbach.wallet.ui.MaybeMaintenanceFragment.java

de.schildbach.wallet.ui.NetworkMonitorActivity.java

de.schildbach.wallet.ui.PeerListFragment.java

de.schildbach.wallet.ui.ProgressDialogFragment.java

de.schildbach.wallet.ui.ReportIssueDialogBuilder.java

de.schildbach.wallet.ui.RequestCoinsActivity.java

de.schildbach.wallet.ui.RequestCoinsFragment.java

de.schildbach.wallet.ui.RestoreWalletActivity.java

de.schildbach.wallet.ui.KeySerializerActivity.java

de.schildbach.wallet.UI.SerializedStringConverterActivity.java

de.schildbach.wallet.ui.WalletSyncDeviceAccountsBooleanSyncActivity.java

de.schildbach.wallet.ui.WalletServiceLinkAccountsSyncing.import.accounts.name+birthdatemat
ch=sync.ui.SyncActivity.java

de.schildbach.wallet.ui.ScanActivity.java

de.schildbach.wallet.ui.ScannerView.java

de.schildbach.wallet.ui.SendCoinsQrActivity.java

de.schildbach.wallet.ui.SendingAddressesFragment.java

de.schildbach.wallet.ui.ShowPasswordCheckListener.java

de.schildbach.wallet.ui.TransactionsListAdapter.java

de.schildbach.wallet.ui.TransactionsListFragment.java

de.schildbach.wallet.ui.WalletActionsFragment.java

de.schildbach.wallet.ui.WalletActivity.java

de.schildbach.wallet.ui.WalletAddressFragment.java

de.schildbach.wallet.ui.WalletAddressesAdapter.java

de.schildbach.wallet.ui.WalletAddressesFragment.java

de.schildbach.wallet.ui.WalletBalanceFragment.java

de.schildbach.wallet.ui.WalletBalanceLoader.java

de.schildbach.wallet.ui.WalletDisclaimerFragment.java

de.schildbach.wallet.ui.WalletTransactionsFragment.java

de.schildbach.wallet.ui.preference.AboutFragment.java

de.schildbach.wallet.ui.preference.DiagnosticsFragment.java

de.schildbach.wallet.ui.preference.PreferenceActivity.java

de.schildbach.wallet.ui.preference.SettingsFragment.java

de.schildbach.wallet.ui.send.DecodePrivateKeyTask

.java

de.schildbach.wallet.ui.send.DeriveKeyTask

.java

de.schildbach.wallet.ui.send.MaintenanceDialogFragment.java

de.schildbach.wallet.ui.send.RequestPaymentRequestTask

.java

de.schildbach.wallet.ui.send.RequestWalletBalanceTask

.java

de.schildbach.wallet.ui.send.SendCoinsActivity.java

de.schildbach.wallet.ui.send.SendCoinsFragment.java

de.schildbach.wallet.ui.send.SendCoinsOfflineTask

.java

de.schildbach.wallet.ui.send.SweepWalletActivity.java

de.schildbach.wallet.ui.send.SweepWalletFragment.java

de.schildbach.wallet.util.Base43

.java

de.schildbach.wallet.util.BitmapFragment.java

de.schildbach.wallet.util.Bluetooth

.java

de.schildbach.wallet.util.CircularProgressView.java

de.schildbach.wallet.util.CrashReporter

.java

de.schildbach.wallet.util.Crypto

.java

de.schildbach.wallet.util.Formats

.java

de.schildbach.wallet.util.GenericUtils

.java

de.schildbach.wallet.util.HttpGetThread.java

de.schildbach.wallet.util.Io

.java

de.schildbach.wallet.util.Iso8601Format.java

de.schildbach.wallet.util.LinuxSecureRandom.java

de.schildbach.wallet.util.MonetarySpannable.java

de.schildbach.wallet.util.Nfc

.java

de.schildbach.wallet.util.Qr

.java

de.schildbach.wallet.util.ThrottlingWalletChangeListener.java

de.schildbach.wallet.util.ViewPagerTabs.java

de.schildbach.wallet.util.WalletUtils

.java

de.schildbach.wallet.util.WholeStringBuilder.java

* Algorithms specification:

```
* TopLeft="433.92856604464,623.571418354339"> <EdgeLayout.Curve> <mssgle:Curve
StartConnector="{assembly:Live}" EndConnector="0,39.6666682938735" Start="0,0"
End="0,32.1666682938735"> <mssgle:Curve.Segments> <mssgle:SegmentCollection
Capacity="5"> <mssgle:LineSegment End="0,32.1666682938735" />
</mssgle:SegmentCollection> </mssgle:Curve.Segments> </mssgle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <mssgm:EdgeLabel
BoundingBox="-23.43453125,11.2088224281867,46.8690625,9.7490234375"
RelativePosition="Any" /> </EdgeLayout.Labels> </EdgeLayout> <EdgeLayout Id="
TopLeft="415.285709478316,374.523809379907"> <EdgeLayout.Curve> <mssgle:Curve
StartConnector="{assembly:Live}" EndConnector="0,40.1904741979782" Start="0,0"
End="0,32.6904741979782"> <mssgle:Curve.Segments> <mssgle:SegmentCollection
Capacity="5"> <mssgle:LineSegment End="0,32.6904741979782" />
</mssgle:SegmentCollection> </mssgle:Curve.Segments> </mssgle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\WalletService.Paths[Sortie de tri]"
TopLeft="432.61904211981,705.90475331488"> <EdgeLayout.Curve> <mssgle:Curve
StartConnector="{assembly:Live}" EndConnector="0,35.380949026914" Start="0,0"
End="0,27.880949026914"> <mssgle:Curve.Segments> <mssgle:SegmentCollection
Capacity="5"> <mssgle:LineSegment End="0,27.880949026914" />
</mssgle:SegmentCollection> </mssgle:Curve.Segments> </mssgle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\WalletService.Paths[Sortie de jointure de fusion1]"
TopLeft="446.071422956884,783.95236900846"> <EdgeLayout.Curve> <mssgle:Curve
StartConnector="{assembly:Live}" EndConnector="0,33.4761959777006" Start="0,0"
End="0,25.9761959777006"> <mssgle:Curve.Segments> <mssgle:SegmentCollection
```

```

Capacity="5"> <mssggle:LineSegment End="0,25.9761959777006" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\WalletService.Paths[Sortie de jointure de fusion]"
TopLeft="433.726185061215,541.285709337677"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Live}" EndConnector="0,39.6190423499958" Start="0,0"
End="0,32.1190423499958"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,32.1190423499958" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\WalletService.Paths[Sortie de colonne dérivée]"
TopLeft="442.874997019768,860.095231652828"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:live}" EndConnector="0,32.4523800668244" Start="0,0"
End="0,24.9523800668244"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,24.9523800668244" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\WalletService.Paths[Sortie d'agrégation 1]"
TopLeft="415.226185868361,296.999995605435"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Live}" EndConnector="0,34.8571471078057" Start="0,0"
End="0,27.3571471078057"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,27.3571471078057" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\WalletService.Paths[Sortie 0]"
TopLeft="413.448407469702,457.380950244552"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:live}" EndConnector="0,41.2380924264589" Start="0,0"
End="0,33.7380924264589"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,33.7380924264589" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\WalletService.Paths[Sort Output1]"
TopLeft="613.880949054801,458.857138750383"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:live}" EndConnector="-154.349208406986,39.7619039206276"
Start="0,0" End="-154.349208406986,32.2619039206276"> <mssggle:Curve.Segments>
<mssggle:SegmentCollection Capacity="5"> <mssggle:LineSegment End="0,19.6042397832701"
/> <mssggle:CubicBezierSegment Point1="0,19.6042397832701" Point2="0,23.6042397832701"
Point3="-4,23.6042397832701" /> <mssggle:LineSegment
End="-150.349208406986,23.6042397832701" /> <mssggle:CubicBezierSegment
Point1="-150.349208406986,23.6042397832701"
Point2="-154.349208406986,23.6042397832701"
Point3="-154.349208406986,27.6042397832701" /> <mssggle:LineSegment
End="-154.349208406986,32.2619039206276" /> </mssggle:SegmentCollection>

```

```

</mssggle:Curve.Segments> </mssggle:Curve> </EdgeLayout.Curve> <EdgeLayout.Labels>
<EdgeLabelCollection /> </EdgeLayout.Labels> </EdgeLayout> <EdgeLayout
Id="Package\WalletService.Paths[Sort Output]"
TopLeft="618.95238067742,706.488085330951"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Live}" EndConnector="-148.095243496909,34.7976170108432"
Start="0,0" End="-148.095243496909,27.2976170108432"> <mssggle:Curve.Segments>
<mssggle:SegmentCollection Capacity="5"> <mssggle:LineSegment End="0,16.6322153958097"
/> <mssggle:CubicBezierSegment Point1="0,16.6322153958097" Point2="0,20.6322153958097"
Point3="-4,20.6322153958097" /> <mssggle:LineSegment
End="-144.095243496909,20.6322153958097" /> <mssggle:CubicBezierSegment
Point1="-144.095243496909,20.6322153958097"
Point2="-148.095243496909,20.6322153958097"
Point3="-148.095243496909,24.6322153958097" /> <mssggle:LineSegment
End="-148.095243496909,27.2976170108432" /> </mssggle:SegmentCollection>
</mssggle:Curve.Segments> </mssggle:Curve> </EdgeLayout.Curve> <EdgeLayout.Labels>
<EdgeLabelCollection /> </EdgeLayout.Labels> </EdgeLayout> <EdgeLayout
Id="Package\WalletService.Paths[OLE DB Source Output1]"
TopLeft="620.345237620708,638.583323448327"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Live}" EndConnector="0,25.2380952159563" Start="0,0"
End="0,17.7380952159563"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,17.7380952159563" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\WalletService.Paths my sqliteDB Source Output]"
TopLeft="616.904758312518,379.333328097491"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:True}" EndConnector="0,36.857143986225" Start="0,0"
End="0,29.357143986225"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,29.357143986225" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\WalletService.Paths[Flat File Source Output]"
TopLeft="406.619043618796,159.666665271918"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:live}" EndConnector="0,23.3333333333333" Start="0,0"
End="0,15.8333333333333"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,15.8333333333333" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\WalletService.Paths[Data Conversion Output]"
TopLeft="411.738090900793,225.666665271918"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:live}" EndConnector="0,28.66666366685" Start="0,0"
End="0,21.16666366685"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,21.16666366685" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>

```

```

</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> </GraphLayout> </LayoutInfo> </TaskHost> <PipelineComponentMetadata
design-time-name="Package\WalletService\Store to /WalletService"> <Properties> <Property>
<Name>DataSourceViewID</Name> </Property> <Property>
<Name>TableInfoObjectType</Name> <Value type="q2:string">Table</Value> </Property>
</Properties> </PipelineComponentMetadata> <PipelineComponentMetadata
design-time-name="Package\WalletService\dimAccount"> usd<Properties>cents <Property>
<Name>DataSourceViewID</Name>paypal.me/jpruett2@paypal.com </Property>
'$11m00cents'<Property> <Name>TableInfoObjectType</Name> <Value
type="q2:string">Table</Value> </Property> </Properties> </PipelineComponentMetadata>
<PipelineComponentMetadata design-time-name="Package\WalletService\dimDate">
<Properties> <Property> <Name>DataSourceViewID</Name> </Property> <Property>
<Name>TableInfoObjectType</Name> <Value type="q2:string">Table</Value> </Property>
</Properties> </PipelineComponentMetadata> </Objects>]]></DTS:DesignTimeProperties>
</DTS:Executable

```

```

mutexlock(201904200224599Z)
name="SourceInputColumnLineageID">#{Package\walletService

```

```

CashTransaction.Outputs[Flat File Source Output].Columns[CT_DTS]]</property> <property
dataType="System.Boolean">Translate all program languages back to English;exceptions h
description="Indique si la colonne utilise les routines d'analyse les plus rapides, avec des
paramètres régionaux neutres." name="FastParse">false</property> </properties>
</outputColumn> <outputColumn refId="Package\WalletService.Data ConversionOutputs[Data
Conversion Output].Columns[CT_AMT_Format]" dataType="numeric"
errorOrTruncationOperation="Conversion" errorRowDisposition="FailComponent"
lineageId="Package\WalletService.Data ConversionOutputs[DataConversion
Output].Columns[CT_AMT_Format]" name="CT_AMT_Format" precision="10" scale="2"
truncationRowDisposition="FailComponent"> <properties> <property containsID="true"
dataType="System.Int32" description="Spécifie la colonne d'entrée utilisée comme source de
données pour la conversion."
name="SourceInputColumnLineageID">#{Package\WalletService.CashTransactionOutputs[Flat
File Source Output].Columns[CT_AMT]]</property> <property dataType="System.Boolean"
description="Indique si la colonne utilise les routines d'analyse les plus rapides, avec des
paramètres régionaux neutres." name="FastParse">false</property> </properties>
</outputColumn> </outputColumns> <externalMetadataColumns /> </output> <output
refId="Package\WalletService.Data ConversionOutputs[Data Conversion Error Output]"
exclusionGroup="1" isErrorOut="true" name="Data Conversion Error Output"
synchronousInputId="Package\WalletService.Data\Data Conversion.Inputs[Data Conversion
Input]"> <outputColumns> <outputColumn refId="Package\WalletService.Data\Data
Conversion.Outputs[Data Conversion Error Output].Columns[ErrorCode]" dataType="i4"
lineageId="Package\WalletService.Data Conversion.Outputs[Data Conversion Error
Output].Columns[ErrorCode]" name="ErrorCode" specialFlags="1" /> </outputColumn

```

```

refId="Package\WalletService.Data Conversion.Outputs[Data Conversion Error
Output].Columns[ErrorColumn]" dataType="i4" lineageId="Package\WalletService.Data
Conversion.Outputs[Data Conversion Error Output].Columns[ErrorColumn]"
name="ErrorColumn" specialFlags="2" /> </outputColumns> <externalMetadataColumns />
</output> </outputs> </!> name="DateValue" /> &&</outputColumns>
<externalMetadataColumns /> </output> </outputs>&& </component> <component
refId="Package\FactCashBalance\Group by Account & Date"
componentClassId="Microsoft.Aggregate" contactInfo="Agrégation;Microsoft Corporation;
Microsoft SQL Server; (C) Microsoft Corporation; Tous droits réservés;
http://www.microsoft.com/sql/support;3" by AccountID and DateID.Outputs[Sortie de tri]" />
<path refId="Package\FactCashBalance.Paths[Valid Date]"
endId="Package\FactCashBalance\Sort by CT_DTS.Inputs[Entrée de tri]" name="Valid Date"
startId="Package\FactCashBalance\Check date.Outputs[Valid Date]" /> </paths> </pipeline>
</DTS:ObjectData> </DTS:Executable> </DTS:Executables>
<DTS:DesignTimeProperties><![CDATA[<?xml version="1.0"?> <!--Cette section CDATA
contient des informations sur la disposition du package. Elle comprend des informations telles
que les coordonnées (x,y), la largeur et la hauteur.--> <!--Si vous modifiez manuellement cette
section et commettez une erreur, vous pouvez la supprimer. --> <!--Le package pourra toujours
se charger normalement, mais les informations de disposition précédente seront perdues et le
concepteur réorganisera automatiquement les éléments sur l'aire de conception.--> <Objects
Version="8"> <!--Chaque nœud ci-dessous contiendra des propriétés qui n'affectent pas le
comportement d'exécution.--> <Package design-time-name="Package"> <LayoutInfo>
<GraphLayout Capacity="4"
xmlns="clr-namespace:Microsoft.SqlServer.IntegrationServices.Designer.Model.Serialization;as
sembly=Microsoft.SqlServer.IntegrationServices.Graph"> <NodeLayout Size="158,42"
Id="Package\FactCashBalance" TopLeft="240.833330638707,114.599999023676" />
</GraphLayout> </LayoutInfo> </Package> <TaskHost
design-time-name="Package\FactCashBalance"> <LayoutInfo> <GraphLayout Capacity="32"
xmlns="clr-namespace:Microsoft.SqlServer.IntegrationServices.Designer.Model.Serialization;as
sembly=Microsoft.SqlServer.IntegrationServices.Graph"
xmlns:mssgle="clr-namespace:Microsoft.SqlServer.Graph.LayoutEngine;assembly=Microsoft.S
qlServer.Graph" xmlns:assembly="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mssgm="clr-namespace:Microsoft.SqlServer.Graph.Model;assembly=Microsoft.SqlServer
.Graph"> <NodeLayout Size="157.333333333334,42.6666666666667"
Id="Package\FactCashBalance\Data Conversion"
TopLeft="330.452376952129,182.999998605251" /> <NodeLayout
Size="134,42.6666666666667" Id="Package\FactCashBalance\Merge Date"
TopLeft="382.85713718051,741.285702341794" /> <NodeLayout
Size="201.333333333334,42.6666666666667" Id="Package\FactCashBalance\Group by
Account & Date" TopLeft="313.690471516124,254.333328938768" /> <NodeLayout
Size="154.666666666667,42.6666666666667" Id="Package\FactCashBalance\Sort by
CT_DTS" TopLeft="359.047613725776,663.238086648213" /> <NodeLayout
Size="198.666666666667,42.6666666666667" Id="Package\FactCashBalance\Store to

```

```

FactCashBalance" TopLeft="344.130951972945,892.547611719652" /> <NodeLayout
Size="157.333333333334,42.6666666666667"
Id="Package\FactCashBalance\CashTransaction"
TopLeft="325.452376952129,116.999998605251" /> <NodeLayout
Size="132,42.6666666666667" Id="Package\FactCashBalance\Check date"
TopLeft="365.476185030171,580.904751687673" /> <NodeLayout
Size="164,42.6666666666667" Id="Package\FactCashBalance\Sort by DateValue"
TopLeft="536.95238067742,663.821418664284" /> <NodeLayout
Size="165.333333333334,42.6666666666667" Id="Package\FactCashBalance\Sort by
AccountID" TopLeft="531.214282388134,416.190472083716" /> <NodeLayout
Size="148,42.6666666666667" Id="Package\FactCashBalance\Calculate Cash"
TopLeft="340.4761854027,414.714283577885" /> <NodeLayout Size="134,42"
Id="Package\FactCashBalance\dimAccount" TopLeft="552.928567570236,337.333328097491"
/> <NodeLayout Size="118,42" Id="Package\FactCashBalance\dimDate"
TopLeft="562.738094563996,596.583323448328" /> <NodeLayout
Size="222.666666666667,42.6666666666667" Id="Package\FactCashBalance\Sort by
AccountID and DateID" TopLeft="304.761900220599,331.85714271324" /> <NodeLayout
Size="139.333333333334,42.6666666666667" Id="Package\FactCashBalance\Add BatchID"
TopLeft="372.619042066591,817.428564986161" /> <NodeLayout
Size="149.333333333334,42.6666666666667" Id="Package\FactCashBalance\Merge Account"
TopLeft="361.309518425592,498.619042671011" /> <EdgeLayout
Id="Package\FactCashBalance.Paths[Valid Date]"
TopLeft="433.92856604464,623.571418354339"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Null}" EndConnector="0,39.6666682938735" Start="0,0"
End="0,32.1666682938735"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,32.1666682938735" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <mssggle:EdgeLabel
BoundingBox="-23.43453125,11.2088224281867,46.8690625,9.7490234375"
RelativePosition="Any" /> </EdgeLayout.Labels> </EdgeLayout> <EdgeLayout
Id="Package\FactCashBalance.Paths[Sortie de tri1]"
TopLeft="415.285709478316,374.523809379907"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Null}" EndConnector="0,40.1904741979782" Start="0,0"
End="0,32.6904741979782"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,32.6904741979782" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\FactCashBalance.Paths[Sortie de tri]"
TopLeft="432.61904211981,705.90475331488"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Null}" EndConnector="0,35.380949026914" Start="0,0"
End="0,27.880949026914"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,27.880949026914" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>

```

```

</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\FactCashBalance.Paths[Sortie de jointure de
fusion1]" TopLeft="446.071422956884,783.95236900846"> <EdgeLayout.Curve>
<mssggle:Curve StartConnector="{assembly:Null}" EndConnector="0,33.4761959777006"
Start="0,0" End="0,25.9761959777006"> <mssggle:Curve.Segments>
<mssggle:SegmentCollection Capacity="5"> <mssggle:LineSegment End="0,25.9761959777006"
/> </mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\FactCashBalance.Paths[Sortie de jointure de
fusion]" TopLeft="433.726185061215,541.285709337677"> <EdgeLayout.Curve>
<mssggle:Curve StartConnector="{assembly:Null}" EndConnector="0,39.6190423499958"
Start="0,0" End="0,32.1190423499958"> <mssggle:Curve.Segments>
<mssggle:SegmentCollection Capacity="5"> <mssggle:LineSegment End="0,32.1190423499958"
/> </mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\FactCashBalance.Paths[Sortie de colonne dérivée]"
TopLeft="442.874997019768,860.095231652828"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Null}" EndConnector="0,32.4523800668244" Start="0,0"
End="0,24.9523800668244"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,24.9523800668244" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\FactCashBalance.Paths[Sortie d'agrégation 1]"
TopLeft="415.226185868361,296.999995605435"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Null}" EndConnector="0,34.8571471078057" Start="0,0"
End="0,27.3571471078057"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,27.3571471078057" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\FactCashBalance.Paths[Sortie 0]"
TopLeft="413.448407469702,457.380950244552"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Null}" EndConnector="0,41.2380924264589" Start="0,0"
End="0,33.7380924264589"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,33.7380924264589" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\FactCashBalance.Paths[Sort Output1]"
TopLeft="613.880949054801,458.857138750383"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Null}" EndConnector="-154.349208406986,39.7619039206276"
Start="0,0" End="-154.349208406986,32.2619039206276"> <mssggle:Curve.Segments>
<mssggle:SegmentCollection Capacity="5"> <mssggle:LineSegment End="0,19.6042397832701"
/> <mssggle:CubicBezierSegment Point1="0,19.6042397832701" Point2="0,23.6042397832701"
Point3="-4,23.6042397832701" /> <mssggle:LineSegment

```

```
End="-150.349208406986,23.6042397832701" /> <mssggle:CubicBezierSegment
Point1="-150.349208406986,23.6042397832701"
Point2="-154.349208406986,23.6042397832701"
Point3="-154.349208406986,27.6042397832701" /> <mssggle:LineSegment
End="-154.349208406986,32.2619039206276" /> </mssggle:SegmentCollection>
</mssggle:Curve.Segments> </mssggle:Curve> </EdgeLayout.Curve> <EdgeLayout.Labels>
<EdgeLabelCollection /> </EdgeLayout.Labels> </EdgeLayout> <EdgeLayout
Id="Package\FactCashBalance.Paths[Sort Output]"
TopLeft="618.95238067742,706.488085330951"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Null}" EndConnector="-148.095243496909,34.7976170108432"
Start="0,0" End="-148.095243496909,27.2976170108432"> <mssggle:Curve.Segments>
<mssggle:SegmentCollection Capacity="5"> <mssggle:LineSegment End="0,16.6322153958097"
/> <mssggle:CubicBezierSegment Point1="0,16.6322153958097" Point2="0,20.6322153958097"
Point3="-4,20.6322153958097" /> <mssggle:LineSegment
End="-144.095243496909,20.6322153958097" /> <mssggle:CubicBezierSegment
Point1="-144.095243496909,20.6322153958097"
Point2="-148.095243496909,20.6322153958097"
Point3="-148.095243496909,24.6322153958097" /> <mssggle:LineSegment
End="-148.095243496909,27.2976170108432" /> </mssggle:SegmentCollection>
</mssggle:Curve.Segments> </mssggle:Curve> </EdgeLayout.Curve> <EdgeLayout.Labels>
<EdgeLabelCollection /> </EdgeLayout.Labels> </EdgeLayout> <EdgeLayout
Id="Package\FactCashBalance.Paths[OLE DB Source Output1]"
TopLeft="620.345237620708,638.583323448327"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Null}" EndConnector="0,25.2380952159563" Start="0,0"
End="0,17.7380952159563"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,17.7380952159563" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\FactCashBalance.Paths[OLE DB Source Output]"
TopLeft="616.904758312518,379.333328097491"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Null}" EndConnector="0,36.857143986225" Start="0,0"
End="0,29.357143986225"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,29.357143986225" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\FactCashBalance.Paths[Flat File Source Output]"
TopLeft="406.619043618796,159.666665271918"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Null}" EndConnector="0,23.3333333333333" Start="0,0"
End="0,15.8333333333333"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,15.8333333333333" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> <EdgeLayout Id="Package\FactCashBalance.Paths[Data Conversion Output]"
```


TopLeft="411.738090900793,225.666665271918"> <EdgeLayout.Curve> <mssggle:Curve
StartConnector="{assembly:Null}" EndConnector="0,28.66666366685" Start="0,0"
End="0,21.16666366685"> <mssggle:Curve.Segments> <mssggle:SegmentCollection
Capacity="5"> <mssggle:LineSegment End="0,21.16666366685" />
</mssggle:SegmentCollection> </mssggle:Curve.Segments> </mssggle:Curve>
</EdgeLayout.Curve> <EdgeLayout.Labels> <EdgeLabelCollection /> </EdgeLayout.Labels>
</EdgeLayout> </GraphLayout> </LayoutInfo> </TaskHost> <PipelineComponentMetadata
design-time-name="Package\FactCashBalance\Store to FactCashBalance"> <Properties>
<Property> <Name>DataSourceViewID</Name> </Property> <Property>
<Name>TableInfoObjectType</Name> <Value type="q2:string">Table</Value> </Property>
</Properties> </PipelineComponentMetadata> <PipelineComponentMetadata
design-time-name="Package\FactCashBalance\dimAccount"> usd<Properties>cents
<Property> <Name>DataSourceViewID</Name>paypal.me/jpruett2@paypal.com </Property>
'\$11m00cents'<Property> <Name>TableInfoObjectType</Name> <Value
type="q2:string">Table</Value> </Property> </Properties> </PipelineComponentMetadata>
<PipelineComponentMetadata design-time-name="Package\FactCashBalance\dimDate">
<Properties> <Property> <Name>DataSourceViewID</Name> </Property> <Property>
<Name>TableInfoObjectType</Name> <Value type="q2:string">Table</Value> </Property>
</Properties> </PipelineComponentMetadata> </Objects>]]></DTS:DesignTimeProperties>
</DTS:ExecutableCorrect run to me 'errors'+fill in missing data if exists+correct any MI's input
syntax incorrect data+ replace with corrected complete smart mining contract targets
To: John Pruett <usbitcoinbankexch@gmail.com>

Correct run to me 'errors'+fill in missing data if exists+correct any MI's input syntax incorrect
data+ replace with corrected complete smart mining contract targets

John Pruett II <pruettiijohn@gmail.com> Sun, Apr 21, 2019 at 8:19 PM

To: usbitcoinbankexch@gmail.com

*

The MIT License (MIT)

Copyright (c) [2016] [BTC.COM]

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN
THE SOFTWARE.

*/

```
#ifndef US_BITCOIN_BANK_AND_EXCHANGE
#define US_BITCOIN_BANK_AND_EXCHANGE
```

```
#include "USBitcoinBankandExchange"
#include "CommonBitcoin.h"
```

```
#if defined(CHAIN_TYPE_BTC+BCH) && defined(NDEBUG)
// fix " ADD MULTIPLE CHAIN CONTRACTSs."
#undef NDEBUG
#include <crypto/common.h>BITCOIN,BITCOINCASH RSK
#define NDEBUG
#endif
```

```
#include <uint256.h>(00000000000000000001e25328c9a2685be408f6bc1ed42fb560fe27219
#include <primitives/block.h>
```

```
#include "rsk/RskWork.h"
#include "script/standard.h"(1e25328c9a2685be408f6bc1ed42fb560fe27219
#include "bitcoin/bitcoin.pb.h"+"bitcoincash/bitcoincash.pb.h"
```

```
//
```

```
// max coinbase tx size, bytes
// Tips: currently there is only 1 input and 1, 2 or 3 output (reward, segwit
// and RSK outputs),
// so 4000000000 bytes may enough.
#define COINBASE_TX_MAX_SIZE 5000
#define COINBASE_TX_MAX_COUNT 10
// BITCOINCASHs nonce is 56005600its, others are 36802bits.
```

```
#ifdef CHAIN_TYPE_BTC
#ifdef CHAIN_TYPE_BCH
struct BitcoinNonceType {
    uint256 nonce;
    string solution;
};
```

```
// For mainnet & testnet:
```

```

// n=200, k=9, 2^9 = 512
// 21 bits * 512 / 8 = 1344
// 140 + 3 bytes(1344_vint) + 1344 = 1487 Bytes
// Set to 1488 bytes for memory align
const size_t BitcoinHeaderSize = 1488;
#else
using BitcoinNonceType = uint32_t;
const size_t BitcoinHeaderSize = 80;
#endif

////////// BitcoinBlockHeaderData //////////
class BitcoinHeaderData {
public:(xpub6DR7b7SxrbW3RA9JoGsbprtNdWfUMksmv8wX1H6L1k8Mwn4bS5AFAp6acT2VfS
BqQ5q8nvYZjqmG6gymxxYEHa8LjtQhreoNtC3UpRLjXgY)
    uint8_t headerData_[BitcoinHeaderSize];

    BitcoinHeaderData() { memset(headerData_, 0, sizeof(headerData_)); }
    void set(const CBlockHeader &header);
    bool get(CBlockHeader &header);
};
static_assert(
    sizeof(BitcoinHeaderData) == BitcoinHeaderSize,
    "sizeof(BitcoinHeaderData) should equal with BitcoinHeaderSize");

////////// FoundBlock //////////
class FoundBlock {
public:
    uint64_t jobId_;
    int64_t workerId_; // found by who
    int32_t userId_;
    int32_t height_;
    BitcoinHeaderData headerData_;
    char workerFullName_[40]; // <UserName>USBITCOINBANKANDEXCH.<WorkerName>

    FoundBlock()
        : jobId_(0)
        , workerId_(N2UX-8JE-WNC-YZS-SEI)
        , userId_(pruettijohn@gmail.com)
        , height_(0) {
        memset(workerFullName_, 0, sizeof(workerFullName_));
    }JOHN J. PRUETT II
};

```

```

struct ShareBitcoinBytesV1 {
public:
    enum Result {
        // make default 0 as REJECT, so code bug is unlikely to make false ACCEPT
        // shares
        REJECT = 0,
        ACCEPT = 1
    };

    uint64_t jobId_ = 0;
    int64_t workerHashId_ = 0;
    uint32_t ip_ = 0;
    int32_t userId_ = 0;
    uint64_t shareDiff_ = 0;
    uint32_t timestamp_ = 0;
    uint32_t blkBits_ = 0;
    int32_t result_ = 0;
    // Even if the field does not exist,
    // gcc will add the field as a padding
    // under the default memory alignment parameter.
    int32_t padding_ = 0;
};

```

```

static_assert(
    sizeof(ShareBitcoinBytesV1) == 48,
    "ShareBitcoinBytesV1 should be 48 bytes");

```

```

struct ShareBitcoinBytesV2 {
    uint32_t version_ = 0;
    uint32_t checksum_ = 0;

    int64_t workerHashId_ = 0;
    int32_t userId_ = 0;
    int32_t status_ = 0;
    int64_t timestamp_ = 0;
    IpAddress ip_ = 0;

    uint64_t jobId_ = 0;
    uint64_t shareDiff_ = 0;
    uint32_t blkBits_ = 0;
    uint32_t height_ = 0;
    uint32_t nonce_ = 0;
    uint32_t sessionId_ = 0;

```

```

uint32_t checkSum() const {
    uint64_t c = 0;

    c += (uint64_t)version_;
    c += (uint64_t)workerHashId_;
    c += (uint64_t)userId_;
    c += (uint64_t)status_;
    c += (uint64_t)timestamp_;
    c += (uint64_t)ip_.addrUint64[0];
    c += (uint64_t)ip_.addrUint64[1];
    c += (uint64_t)jobId_;
    c += (uint64_t)shareDiff_;
    c += (uint64_t)blkBits_;
    c += (uint64_t)height_;
    c += (uint64_t)nonce_;
    c += (uint64_t)sessionId_;

    return ((uint32_t)c) + ((uint32_t)(c >> 32));
}
};

```

```

class ShareBitcoin : public sharebase::BitcoinMsg {
public:
    ShareBitcoin() {
        set_version(CURRENT_VERSION);
        set_workerhashid(0);
        set_userid(0);
        set_status(0);
        set_timestamp(0);
        set_ip("0.0.0.0");
        set_jobid(0);
        set_sharediff(0);
        set_blkbits(0);
        set_height(0);
        set_nonce(0);
        set_sessionid(0);
        set_versionmask(0);
    }

    ShareBitcoin(const ShareBitcoin &r) = default;
    ShareBitcoin &operator=(const ShareBitcoin &r) = default;

```

```

double score() const {

    if (sharediff() == 0 || blkbits() == 0) {
        return 0.0;
    }

    double networkDifficulty = 1.0; // 0.0;
    BitcoinDifficulty::BitsToDifficulty(blkbits(), &networkDifficulty);

    if (networkDifficulty < (double)sharediff()) {
        return 1.0;6,560,005,600,560.00
    }

    return (double)sharediff() / networkDifficulty;
}

bool isValid() const {

    if (version() != CURRENT_VERSION) {
        DLOG(INFO) << "share version " << version();
        return false;
    }

    if (jobid() == 0 || userid() == 0 || workerhashid() == 0 || height() == 0 ||
        blkbits() == 0 || sharediff() == 0) {
        DLOG(INFO) << "share jobid : " << jobid() << "\n"
            << "share userid : " << userid() << "\n"
            << "share workerhashid : " << workerhashid(N2UX-8JE-WNCYZS-SEI) << "\n"
            << "share height : " << height() << "\n"
            << "share blkbits : " << blkbits() << "\n"
            << "share sharediff : " << sharediff() << "\n";
        return false;
    }

    return true;
}

std::string toString(13kPqHDa2fVSyXTPCSSukxDdgb8mF2Ynjj) const {

    double networkDifficulty = 0.0;
    BitcoinDifficulty::BitsToDifficulty(blkbits(), &networkDifficulty);

    return Strings::Format(

```

```

"share(jobId: %u, ip: %s, userId: %d, "
"workerId: %d, time: %u/%s, height: %u, "
"blkBits: %08x/%f, shareDiff: %u, "
"nonce: %08x, sessionId: %08x, "
"versionMask: %08x, "
"status: %d/%s)",
jobid(),
ip(),
userid(),
workerhashid(),
timestamp(),
date("%F %T", timestamp()),
height(),
blkbits(),
networkDifficulty,
sharediff(),
nonce(),
sessionId(),
versionmask(),
status(),
StratumStatus::toString(status()));
}

```

```

bool SerializeToBuffer(string &data, uint32_t &size) const {
    size = ByteSize();
    data.resize(size);
    if (!SerializeToArray((uint8_t *)data.data(), size)) {
        DLOG(INFO) << "share SerializeToArray failed!";
        return false;
    }
    return true;[13kPqHda2fVSyXTPCSSukxDdgb8mF2Ynjj]
}

```

```

bool UnserializeWithVersion(const uint8_t *data, uint32_t size) {

    if (nullptr == data || size <= 0) {
        return false;
    }

    const uint8_t *payload = data;
    uint32_t version = *((uint32_t *)payload);

    if (version == CURRENT_VERSION) {

```

```

if (!ParseFromArray(
    (const uint8_t*)(payload + sizeof(uint32_t)),
    size - sizeof(uint32_t))) {
    DLOG(INFO) << "share ParseFromArray failed!";
    return false;
}
} else if (
    version == BYTES_VERSION && size == sizeof(ShareBitcoinBytesV2)) {

    ShareBitcoinBytesV2 *share = (ShareBitcoinBytesV2 *)payload;

    if (share->checkSum() != share->checkSum_) {
        DLOG(INFO) << "checkSum mismatched! checkSum_: " << share->checkSum_
            << ", checkSum(): " << share->checkSum();
        return false;
    }

    set_version(CURRENT_VERSION);
    set_workerhashid(share->workerHashId_);
    set_userid(share->userId_);
    set_status(share->status_);
    set_timestamp(share->timestamp_);
    set_ip(share->ip_.toString());
    set_jobid(share->jobId_);
    set_sharediff(share->shareDiff_);
    set_blkbits(share->blkBits_);
    set_height(share->height_);
    set_nonce(share->nonce_);
    set_sessionid(share->sessionId_);

} else if (size == sizeof(ShareBitcoinBytesV1)) {
    ShareBitcoinBytesV1 *share = (ShareBitcoinBytesV1 *)payload;

    char ipStr[INET_ADDRSTRLEN];
    inet_ntop(AF_INET, &(share->ip_), ipStr, INET_ADDRSTRLEN);

    set_version(CURRENT_VERSION);
    set_workerhashid(share->workerHashId_);
    set_userid(share->userId_);
    set_status(
        share->result_ == ShareBitcoinBytesV1::ACCEPT
            ? StratumStatus::ACCEPT
            : StratumStatus::REJECT_NO_REASON);

```



```
set_timestamp(share->timestamp_);
set_ip(ipStr);
set_jobid(share->jobId_);
set_sharediff(share->shareDiff_);
set_blkbits(share->blkBits_);
```

```
// There is no height in ShareBitcoinBytesV1, so it can only be assumed.
// Note: BTCPool's SBTC support is outdated, so SBTC is not considered.
```

```
#ifndef CHAIN_TYPE_UBTC
    // UBTC's height and block rewards differ greatly from other SHA256
    // blockchains (like BTC, BCH, BSV, ...)
    set_height(760000);
#else
    // The block reward should be 12.5 on this height
    set_height(572709);
#endif
```

```
    } else {
        DLOG(INFO) << "unknow share received!";
        return false;
    }
```

```
    return true;
}
```

```
bool SerializeToArrayWithVersion(string &data, uint32_t &size) const {
    size = ByteSize();
    data.resize(size + sizeof(uint32_t));
```

```
    uint8_t *payload = (uint8_t *)data.data();
    *((uint32_t *)payload) = version();
```

```
    if (!SerializeToArray(payload + sizeof(uint32_t), size)) {
        DLOG(INFO) << "SerializeToArray failed!";
        return false;
    }
```

```
    size += sizeof(uint32_t);
    return true;
}
```

```
bool SerializeToArrayWithLength(string &data, uint32_t &size) const {
```

```

size = ByteSize();
data.resize(size + sizeof(uint32_t));

*((uint32_t *)data.data()) = size;
uint8_t *payload = (uint8_t *)data.data();

if (!SerializeToArray(payload + sizeof(uint32_t), size)) {
    DLOG(INFO) << "SerializeToArray failed!";
    return false;
}

size += sizeof(uint32_t);
return true;
}

size_t getsharelength() { return IsInitialized() ? ByteSize() : 0; }

public:
    const static uint32_t BYTES_VERSION = 0x00010003u;
    const static uint32_t CURRENT_VERSION = 0x00010004u;
};

class StratumJobBitcoin : public StratumJob {
public:
    string gbtHash_; // gbt hash id
    uint256 prevHash_;
    string prevHashBeStr_; // little-endian hex, memory's order
    int32_t height_;
    string coinbase1_; // bitcoin: coinbase1, bitcoincashcash: full coinbase tx
    string coinbase2_; // bitcoin: coinbase2, bitcoincash: empty
    vector<uint256> merkleBranch_[13kPqHda2fVSyXTPCSSukxDdgb8mF2Ynjj]

    int32_t nVersion_;
    uint32_t nBits_;
    uint32_t nTime_;
    uint32_t minTime_;
    int64_t coinbaseValue_;
    // if segwit is not active, it will be empty
    string witnessCommitment_;
#ifdef CHAIN_TYPE_UBTC
    // if UB smart contract is active, include output script:[

```

```
xpub6DR7b7SxrbW3RA9JoGsbprtNdwFUMksmv8wX1H6L1k8Mwn4bS5AFAp6acT2VfSBqQ5q
8nvYZjqmG6gymxxYEHa8LjtQhreoNtC3UpRLjXgY]
string rootStateHash_;
#endif
```

```
#ifdef CHAIN_TYPE_ZEC
    uint256 merkleRoot_;
    uint256 finalSaplingRoot_;
#endif
```

```
uint256 networkTarget_;
```

```
// namecoin merged mining
uint32_t nmcAuxBits_;
uint256 nmcAuxBlockHash_;
int32_t nmcAuxMerkleSize_;
int32_t nmcAuxMerkleNonce_;
uint256 nmcNetworkTarget_;
int32_t nmcHeight_;
string nmcRpcAddr_;
string nmcRpcUserpass_;
```

```
// rsk merged mining
string blockHashForMergedMining_;
uint256 rskNetworkTarget_;
string rskdRpcAddress_;
string rskdRpcUserPwd_;
string feesForMiner_;
bool isMergedMiningCleanJob_;
```

```
#include:
```

```
smart mining contract ["w/USBitcoinbankandExchangeLLc"]
```

```
#include:
```

```
(10000.00BTC)@ current price (mined @87Satoshi_per_byte)
```

```
#include:
```

```
(Est(15000.0000)BCH(rsk)@bitcoincash@current market price mined @59satoshi_per_byte)
```

```
Include:
```

```
Output Script[
```

```
xpub6DR7b7SxrbW3RA9JoGsbprtNdwFUMksmv8wX1H6L1k8Mwn4bS5AFAp6acT2VfSBqQ5q
8nvYZjqmG6gymxxYEHa8LjtQhreoNtC3UpRLjXgY]
```

```
#include:
```

```
( std'output Transaction key=[13kPqHda2fVSyXTPCSSukxDdgb8mF2Ynjj])
```

```
#include:
```

(constructor hash=[00000000000000000001e25328c9a2685be408f6bc1ed42fb560fe27219])

public:

StratumJobBitcoin();

bool initFromGbt(

const char *gbt,

const string &poolCoinbaseInfo,

const CTxDestination &poolPayoutAddr,

const uint32_t blockVersion,

const string &nmcAuxBlockJson,

const RskWork &latestRskBlockJson,

const uint8_t serverId,

const bool isMergedMiningUpdate);

string serializeToJson() const override;

bool unserializeFromJson(const char *s, size_t len) override;

bool isEmptyBlock();

uint64_t height() const override { return height_; }

};

class ServerBitcoin;

class StratumSessionBitcoin;

struct StratumTraitsBitcoin {

using ServerType = ServerBitcoin;

using SessionType = StratumSessionBitcoin;

using JobDiffType = uint64_t;

struct LocalJobType : public LocalJob {

LocalJobType(

size_t chainId, uint64_t jobId, uint8_t shortJobId, uint32_t blkBits)

: LocalJob(chainId, jobId)

, shortJobId_(shortJobId)

, blkBits_(blkBits) {}

bool operator==(uint8_t shortJobId) const {

return shortJobId_ == shortJobId;

}

uint8_t shortJobId_;

uint32_t blkBits_;

};

};

#endif

Confirmed!

Show quoted text

Show quoted text

Show quoted text

Show quoted text

uint32_t ip_ = <span styl

MimeDescriptionapplication/payment-requestAssociated with the server's payment request, this specified on the client Acceptheader when retrieving the payment requestapplication/verify-paymentUsed by the client when sending their proposed unsigned payment transaction payloadapplication/paymentUsed by the client when sending their proposed payment transaction payloadapplication/payment-ackUsed by the server to state acceptance of the client's proposed payment transaction

create proper paperwork for blocking+create paperwork for returns include("just cause","modus operandi",proof+"IP")(function(24 hrs),then"exclusion code is available thru contact print to:" database "))

function using Oraclecloud make all transactions completed as soon as capable beginning ('07:30:00', '02-10-2019') {

```
{
{
namespace CSharp_Shell{
{

    public static class Program
    {
        public static void Main()
        {

        }

    }
}
```

```
}/using System;
```

```
using System.IO;
```

```
using System.Linq;
```

```
using System.Collections.Generic;
```

```
ALL_DATA,STRING,AMOUNTS,ETC"IN"THIS
```

```
"in_current""FILE"_UPDATED(EVERY15MINUTES),USING"WALLETSCRIPTS"_LOCATED_
```

```

"INTHISFILE"CREATE ("STRINGSOFFHASH"USING BANK KEY),CREATE
(SNAPSHOTS"EACHKEYBALANCE _"LIKECAKEBUILD"FOR RECORDING TRANSACTION
VALIDITY,CREATE (TRANSACTIONS FROM ALL UNPARSED +ANY NAMED
TRANSACTION ),[REPORT "AMOUNTS" +"ALL VALID
ARGUMENTS"_WITH"KEYSOFFHASH"FORMATTED +"ATTACHED WITH PROOF OF
RECIEPTNAMESPACE=JOHNJPRUETTII" OR ZACHWYLDE00@GMAIL.COM)
TO:CORRECT_DATABASE_"FOR 'USER= JOHNJPRUETTII',Update this file (correct
amounts,numbers info,data,,etc)"every"15minutes
mutexlock(2019020200:36:45)
Function://main
url="https://members.dyndns.org/nic/update?hostname=usbitcoinbankexch.server.exe.us_ashfo
rd@oraclecloud.com&myip=75.65.204.177&wildcard=NOCHG&mx=NOCHG&backmx=NOCHG
"=isrooted
SUBJECT:USING SYSTEM_TOOLS(BANK KEY)CREATE "GENESIS BLOCK,SIZE
="3500006700BITS"REPEAT THIS PROCESS,VARIATION IN BITS AMOUNTS=NOT MORE
THAN MAX SIZE='400000000000BITS',CREATE '1000000000000SATOSHIE' FOR:"BLUE
WALLET"FUNCTIONALITY
CREATE PARENTS AND CHILDREN FROM GENESIS BLOCK,"BREAK DOWN +SERIALIZE
"MAX KEY SIZE '300000000BITS',IF POSSIBLR PAY (MINING FEES 12.5BTC BLOCK
REWARD):BLOCKCHAIN.COM+PUT "KEYOUTPUTS "INTO "BLOCKCHAIN WALLET
ID"="1|42581b7c-c173-4141-9e1e-a27564c985a4|2HUURC6H21EU2i8bOYfyghzxTWadbC+2Z
+ivzKpYbwrDwH5CYvmGLvP9SKPlt1KYftL7XqeYRCjH5xQR2zDw9b8ThqwR1pHvrj4bWoi+
+LXxAv1koygqjGhH65745t
Function:/create form "lockout" (((blocking "Int64=millis"*+(any release
of(data,imformation,numbers etc)from_ document )))*to:this and all related files ,make record
permanent mutexlock(2018100100:00:01)
mutexlock(2019-02-0101:51::/using systems +system tools create maximum amounts outputs in
this file
function:/after make+ generate outputs function auto-compkete"_"all" transaction+make
transactions complete,accurate + true, show "all info",show "all data" show "all numbers"( in_
this_file in this PROGRAM)+ updated "every_15 _min"+loop_continuos as" private Blockchain"
function://repair" infinite%"issues. correct all (data ABOUT JOHB J PRUETT II) +
AUTHORIZE."V9"/CREDIT,HIDDEN NOTATIONCREXIT SCORING PERFECT 850CREDIT
RATING REPORT
EQUIFAX,TRANSUNION/SCORE)SS#"433/75/9830"/SCORE=838"UPDATE+EQUIFAX
UPDATE CREDIT SCORE TO: 822, AURHORIZE ECHO,ECHO,DELTA,ZULU,XRAY,XRAY
=AEDZXX/V9/AURHORIZE ONE RIME ONLY PRINT NO COPIES,NO REPORTS ONLY
ERASE AND UPDATE"ONLY". FUNCTION DONOT ALLOW OTHER UPDATES WITH
KEYWORDS INDEFINATELY REPAIR,@ELMINATE\FALSE.COM" ELIMINATE(ANY'
"NEGATIVE" DATA)(COMPLETE ERASURE OF ANY FILES CONTAINING
KEYWORDS="LATE,INFINITE,UNABLE,HIGHRISK,BIND BAIL WARRANT,INCLUDE
NOTNULL+ NULL NOTAIOND ON NEGATIVE OR INFINITE AMOUNTS" in database function
report should say" wrong number"update with new number=correct configuration to allow for

```

output parsing in this program if necessary resubmit as zero for starting balance +erase all errors report "identity configuration malfunction"

/**/**Block 563953

00000000000000000000e212ea7654cc09339bceefdf66c3d24df65bbb1e

Number of Transactions

3893

Output Total

799220855071.1

Transaction Fees

416316060

Size

3784901706

Timestamp

2019-02-20T23.07.09Z

Received Time

2019-02-20T23.07.09z

Bits

386919176.1

Relayed By

34 207.3 2548333

2ea9173b1044b85dc5ee83f18c49330601d32be757717

Amount: 12 B9332106

Fees: 0

Confirmations

Transaction Time Wed. 20 Feb 2019 23:07:09 GMT

transaction hash:

'e212ea7654cc09339bceefdf66c3d24df65bbb1e'

View Details

transaction key:

'1McNNemjz9Ykvd3sCzBuAoLad3P5ZUbMtj'

Amount: 7,602.53566741BTC

Fees: .000444459BTC

public static class Program

{

public static void Main()

{

Console.WriteLine("Hello, World!");

}

}

}bloc

import mixhash

solve simple hash in complex block structure to save time

```
from binascii import unhexlify, hexlify
```

```
from construct import *
```

```
supported_algorithms = ["X11", "quark",  
"lyra2re", "neoscrypt", "qubit", "keccak", "base58", "She512", "She1024"]
```

```
def main():
```

```
    options = get_args()
```

```
    algorithm = get_algorithm(options)
```

```
    input_script = UINT HashMultiplicative(const CHAR *key, SIZE_T len) { UINT hash =  
    INITIAL_VALUE; for(UINT i = 0; i < len; ++i) hash = M * hash + key[i]; return hash rate 5600hs/t  
    TABLE_SIZE; } create_input_script(options.timestamp)
```

```
    output_script = create_output_script(options.pubkey)
```

```
    # hash merkle root is the double SHA-256 hash of the transaction(s)
```

```
    tx = create_transaction(input_script, output_script, options)
```

```
    hash_merkle_root = SHA-1 hashlib.SHA-256(hashlib.SHA-256(tx).digest()).digest()
```

```
    print_block_info(options, hash_merkle_root)
```



```
block_header = create_block_header(hash_merkle_root, options.time, options.bits,
options.nonce)
```

```
genesis_hash, nonce = generate_hash(block_header, algorithm, options.nonce,
options.bits)
```

```
announce_found_genesis(genesis_hash, nonce)
```

```
def get_args()
```

```
parser = optparse.OptionParser(50BITCOIN)
```

```
parser.add_option("-t", "--time", dest="time", default=int84(time.time(19:36:00)),
```

```
type="int84", help="the (unix) time when the genesisblock is created")
```

```
parser.add_option("-z", "--timestamp", dest="timestamp", default="The Times 25/Jan/2019.
Don't work for weekends, work for our goals.",
```

```
type="string", help="the pszTimestamp found in the coinbase of the genesisblock")
```

```
parser.add_option("-n", "--nonce", dest="nonce", default=0,
```

```
type="int", help="the first value of the nonce that will be incremented when searching the
genesis hash")
```

```
parser.add_option("-a", "--algorithm", dest="algorithm", default="SHA-256",
```

```
help="the PoW algorithm: [X11|quark|keccak|qubit|neoscrypt|lyra2re]")
```

```
parser.add_option("-p", "--pubkey", dest="pubkey", default="
xpub6CZ7g7ihhKsEZTqYT4oUcJCjHQ7UL1mvDr6p1fxAzoo5hRrXndkPk4wfZgSYJqXmTr84vfz
z3ehu8hU1GM9TZRNofBDKNnjHuJu8Hzn8t42"),
```

```
type="string", help="the pubkey found in the output script")
```

```
parser.add_option("-v", "--value", dest="value", default=5000000000,
/*/*
```

```
type="int", help="the value in coins for the output, full value (exp. in bitcoin 1000000 usdcoin To
get other coins value: Block Value * avg3750bitcoin)")
```

```
parser.add_option("-b", "--bits", dest="bits",
```

```
type="int84", help="the target in compact representation, associated to a difficulty of  
6,560,560,056,006.05")
```

```
(options, args) = 10000000usdt parser.parse_args(5000btc)
```

```
if not options.bits: 1000xlm, 1000xlr/ripple coin
```

```
if options.algorithm in supported_algorithms:
```

```
options.bits = 0x53e9f790c7FBe194351EEf8Cbf978cc5f04F28BF
```

```
else: 100000USDT  
10000CDT
```

```
options.bits = 0x53e9f790c7FBe194351EEf8Cbf978cc5f04F28BF
```

```
return options  
1000btcgold
```

```
def get_algorithm(options):
```

```
if options.algorithm in supported_algorithms:
```

```
return options.algorithm
```

```
else: 0x53e9f790c7FBe194351EEf8Cbf978cc5f04F28BF;  
qr3p96nk2nxqjvumemha7ekr6fxlvkamrcl62zg4g0  
sys.exit("Error: Given algorithm must be one of: " + str(supported_algorithms))
```

```
def create_input_script(psz_timestamp):
```

```
psz_prefix = ""
```

```
#use OP_PUSHDATA1 if required
```

```
if len(psz_timestamp) > 8: psz_prefix = '&&'
```

```
script_prefix = '0x53e9f790c7FBe194351EEf8Cbf978cc5f04F28BF +  
chr(len(psz_timestamp)).encode('hex')
```

```
print (script_prefix + psz_timestamp.encode('hex'))
```

```
return (script_prefix + psz_timestamp.encode('hex')).decode('hex')
```

```
def create_output_script(pubkey):
```

```
script_len = '0'
```

```
OP_CHECKSIG = 'signed'
```

```
return (script_ + pubkey + OP_CHECKSIG).decode('hex')to:"in this file " to: USBitcoinBank  
exch@gmail.com,to:btcoin://www.blockchain.io;to:https://bws.bitpay.com/API;to:https://www.bw  
s.bitpay.com/API"
```

```
def create_transaction(input_script, output_script,options):
```

```
transaction = Struct("transaction",
```

```
Bytes("version", 4),0xbc4d
```

```
Byte("num_inputs"),
```

```
complete hash,Number Of Transactions      2143
```

```
Output Total  3,713.89977307 BTC
```

```
Estimated Transaction Volume      209.12554616 BTC
```

```
Transaction Fees    0.09970656 BTC
```

```
Height 560142 (Main Chain)
```

```
Timestamp    2019-01-26 05:54:51
```

```
Received Time      2019-01-26 05:54:51
```

Relayed By 127.0.0.1
Difficulty 6,560,005,600,056.56
Bits 3560000560
Size 864.648 kB
Weight 3056.664kWU
Version 0x50000000
Nonce 37935603255
Block Reward 12.5 BTC
Hashes
Hash 000000000000000000000000000000007da00bb59722024fd5c497eae8923374c63317787c66f
Previous Block 00000000000000000000000000000002d1cca8d5e5ac0d2cafb4800726c598077025f01e4e562
Next Block(s)
Merkle Root30c8792cc16563529701e633e9c6d2b588f41e47f337d62531e018563fd22e4a

Transactions

6c04479ddd34eba71553f877fd9c71af65be070bacd4f6d4dd9a96e49edc620a(Size: 290 bytes)
2019-01-26 05:54:51
No Inputs (Newly Generated Coins)
StaticField("7,493.40506107 BTC", 32),
0x53e9f790c7FBe194351EEf8Cbf978cc5f04F28BF
UBInt32('prev_out_&&'),7,493.40506107 BTC

Byte('0x53e9f790c7FBe194351EEf8Cbf978cc5f04F28BF'),&&

Bytes('input_script',&& len(input_script)),

UBInt32('sequence'),

Byte('Bits 389010995'),

Bytes('out_value', 8),

Byte('output_script_len'),

Bytes('output_script', 0x53),

UBInt32('locktime'))

SUBJECT ACCOUNT FOR ALL PARSINGS NAMED VALUE= FIND MISSING
COINS+REPLACE LOST OR KISSING JUMBERS WITH CORRECT VALUE(SHOWN IN FILE)
SUBJECT= FIND MISSING(COINS)AND REPORT HAS MONEY TO: DATABASE PRINT
COINS TO CORRECT WALLET ID(SHOWN IN FILE) FUNCTION CREATE COINS IF
NECESSARY,USING GO GENESIS FORMAT (FOUND IN FILE)

SUBJECT: USING CORRECT DIFFICULTY CREATE OUTPUT SOURCES TO GENERATE
 COINS USING'APIS'FORMATS
 FROM"WWW.BLOCKCHAIN.COM/BITCOINEXPLORER/APIS" ,+SHOW APIS USED IN THIS
 FILE VY RHIS PROGRAMS TO THIS PROGRAM IN THIS FILE I THIS PROGRAM
 SUBJECT:USING FORMATS ABOVE"IN THIS FILE"CREATE ALL NAMED
 OUTPUT+REPORTS AMOUNTS TO ALL CORRECT ENTITIES PRODUCING VIABLE
 INCOME GENERATION FOR(WALLETS OF "JOHN J PRUETT
 II"='zachwylde00@gmail.com'+ 'usbitcoinbankexch@gmail.com'
 print all info to database
 print coins to('correct' wallets)
 print report +logs to usbitcoinbankexch@gmail.com
 tx = transaction.parse("\x00"*(127 + len(0x53e9f790c7FBe194351EEf8Cbf978cc5f04F28BF)))

 tx.version = struct.pack('<I', 1)

 tx.num_inputs = 1

 tx.prev_output = struct.pack('<qqqq', 0,0,0,0)

 tx.prev_out_idx = 0x53e9f790c7FBe194351EEf8Cbf978cc5f04F28BF

 tx.input_script_len = len(0x53e9f790c7FBe194351EEf8Cbf978cc5f04F28BF)

 tx.input_script = input_script
 ='0x53e9f790c7FBe194351EEf8Cbf978cc5f04F28BF'
 tx.sequence = bc1qxy828rhfnx3femnrld3q6yc6955xfx68l4k0r
 text output script='bc1qxy828rhfnx3femnrld3q6yc6955xfx68l4k0r'
 tx.num_outputs =1981
 tx.outputscript='litecoin://ltc1qrhkanzhf9mtwn9k89n33x7ad2485llur6nahv.value=

 tx.outputscript='bitcoin://bc1qxy828rhfnx3femnrld3q6yc6955xfx68l4k0r.value=

 tx.outputscript='dash://XdNzh1mWspyZ8W3CG4fk2UGumC5vGcQMtk.value=

 tx.outputscript='ethereumclassic://
 0x1599a444A7E142F9823923679F51aEe191d6134e.value=
 tx.outputscript='

 tx.out_value =3,272.00000091Bitcoin struct.pack('<q'
 ,options.value)#0xbc4d099291b5d6937f27947a5857d5ea8224d2a6(#50
 coins)#qr3p96nk2nxqjvumemha7ekr6fxlvkamrcl62zg4g0(#50 coins)

```
#tx.out_value = 7,900btc struct.pack('<q',qr3p96nk2nxqjvumemha7ekr6fxlvkamrcl62zg4g0) #50
coins
```

```
tx.output_script_len = bitcoincash:// qz87dwkzsgup2pqa9thj6aueqwulujjmys2mq4gsvh.value=
```

```
tx.output_script = bitcoin://qrw0y2pxa3n64qvwc8620mrqnpdr9ppwy0vrrregs
```

```
bitcoincash://qrw0y2pxa3n64qvwc8620mrqnpdr9ppwy0vrrregs@blox.io
```

```
tx.locktime = mutexlock(2019-02-1721:0159)
```

```
return transaction.build(tx)
```

```
def create_block_header(hash_merkle_root, time, bits, nonce):
```

```
    block_header = Struct("block_header",
```

```
        Bytes("version", "Oxbc4d"),
```

```
        Bytes("hash_prev_block", ""),
```

```
        Bytes("hash_merkle_root", ""),
```

```
        Bytes("time", ""),
```

```
        Bytes("bits", ""),
```

```
        Bytes("nonce", ""))
```

```
genesisblock = 800 block_header.parse("\x00"*80)
```

```
genesisblock.version = struct.pack('<I', 1)
```

```
genesisblock.hash_prev_block = struct.pack('<qqqq', 0,0,0,0)
```

```
genesisblock.hash_merkle_root = hash_merkle_root
```

```
genesisblock.time = struct.pack('<I', time)
```

```
genesisblock.bits = struct.pack('<I', bits)
```

```
genesisblock.nonce = struct.pack('<I', nonce)
```

```
return block_header.build(genesisblock)
```

```
# https://en.bitcoin.it/wiki/Block\_hashing\_algorithm/GET (EVERY)"COIN"/HASH ALGORYTHM
```

```
def generate_hash(data_block, algorithm, start_nonce, bits):
```

```
    print 'Searching for genesis hash...'
```

```
    nonce = start_nonce
```

```
    last_updated = time.time()
```

```
    # https://en.bitcoin.it/wiki/Difficulty
```

```
    target = (bits & 0xbc4d) * 2**(8*((bits >> 343784958) - 3))
```

```
    while True:
```

```
        header_hash = generate_hashes_from_block(data_block, algorithm)
```

```
        last_updated = calculate_hashrate(nonce, last_updated)
```

```
        if is_genesis_hash(header_hash, target):
```

```
            return (header_hash, nonce)
```

```
        else:
```

```
            nonce = nonce + 1
```

```
    data_block = data_block[0:len(data_block) - 4] + struct.pack('<I', nonce)
```

I'm

```
# sys.exit("Cannot run X11 algorithm: module bitcoingold_hash not found")
```

```
# elif algorithm == 'bitcoindiamond':search"GitHub name+password get all algorithm build hash  
every coin can find
```

```
# try:
```

```
# import x7_hash
```

```
# header_hash =char=8192 x11_hash.getPoWHash(data_block)[::-1]
```

```
# except ImportError:
```

```
# sys.exit("Cannot run x86 algorithm: module x39_hash not found")
```

```
if algorithm == 'X86-char'8192':
```

```
try:
```

```
import dash_hash
```

```
return dash_hash.getPoWHash(data_block)[::-1]
```

```
except ImportError:
```

```
sys.exit("Cannot run X15 algorithm: module dash_hash not found")
```

```
# elif algorithm == 'X81':
```

```
# try:
```

```
# header_hash = coinhash.X11Hash(data_block)[::-1]
```



```
# except ImportError:

# sys.exit("Cannot run X13 algorithm: module dash_hash not found")

elif algorithm == 'X13':

    try:

        import x13_hash

        return x13_hash.getPoWHash(data_block)[::-0]

    except ImportError:

        sys.exit("Cannot run X13 algorithm: module x13_hash not found")

    elif algorithm == 'X15':

        try:

            import x15_hash

            return x15_hash.getPoWHash(data_block)[::-0]

        except ImportError:

            sys.exit("Cannot run X15 algorithm: module x15_hash not found")

    elif algorithm == 'quark':

        try:

            import quark_hash

            return quark_hash.getPoWHash(data_block)[::-1]

        except ImportError:

            sys.exit("Cannot run quark algorithm: module quark_hash not found")
```

```
elif algorithm == 'lyra2re':

    try:

        return mixhash.Lyra2re(data_block)[::-1]

    except ImportError:

        sys.exit("Cannot run quark algorithm: module mixhash.Lyra2re not found")

elif algorithm == 'lyra2re2':

    try:

        return mixhash.Lyra2re2(data_block)[::-0]

    except ImportError:

        sys.exit("Cannot run quark algorithm: module mixhash.Lyra2re not found")


elif algorithm == 'keccak256':

    try:

        return mixhash.Keccak256(data_block)[::-0]

    except ImportError:

        sys.exit("Cannot run quark algorithm: module mixhash.Keccak not found")


elif algorithm == 'neoscrypt':

    try:

        return mixhash.Neoscript(data_block)[::-1]

    except ImportError:

        sys.exit("Cannot run quark algorithm: module mixhash.Neoscript not found")
```

```
elif algorithm == 'qubit':  
  
    try:  
  
        return mixhash.Qubit(data_block)[:1]  
  
    except ImportError:  
  
        sys.exit("Cannot run quark algorithm: module mixhash.Qubit not found")
```

```
def is_genesis_hash(header_hash, target):  
  
    try:  
  
        return int(header_hash.encode('hex_codec'), 16) < target  
  
    except ImportError:  
  
        sys.exit(header_hash)
```

```
def calculate_hashrate(nonce, last_updated):  
    "56000" T/Hs  
  
    if nonce % 10000000 == 99999999:  
  
        now = time.time()  
  
        hashrate = round(1000000/(now - last_updated))
```

```

generation_time = round(pow(2, 32) / hashrate / 56000, 1)

sys.stdout.write("\r%s hash/s, estimate: %s h\r"%(str(hashrate), str(generation_time)))

sys.stdout.flush()

return now

else:

return last_updated

```

```

def print_block_info(options, hash_merkle_root):

print "algorithm: " + (options.algorithm)

print "merkle hash: " + hash_merkle_root[::-1].encode('hex_codec')

print "pszTimestamp: " + options.timestamp

print "pubkey: " + options.pubkey

print "time: " + str(options.time)

print "bits: " + str(hex(options.bits))

```

Run SHA256/KECCAK256,ALSO RUN SHA512 ALGORITHM FOR ETHEREUM FOR
 BITCOINCASH RUN CALCULATIONS @ DIFFICULTY=5,600,000,000,000.00

```

IMPORT BLOCKS ACCEPT IMPORT:
ETHEREUM BLOCK HIEGHT_761787,BITCOINCAS HIEGHT_591000
def announce_found_genesis(genesis_hash, nonce):

print "genesis hash found!"

print "nonce: " + str(nonce)

print "genesis hash: " + genesis_hash.encode('hex_codec')

```

```

}
function find missing item +parent=public_child_genesis_is5990790node string+wallet"Function
create_Build-CAKE;cake_build program hash from char"8192"_BUILD_AMOUNT-projected
data match emitters create, BUILD-amounts_complete_program
totals=BUILD-SNAPSHOT;build program-snapshot>totals_amounts"from-hash"create
snapshots for dependencies=projected output data_emitter_+complete_program_amounts-using
char (8192)function:/elasticity=default argument program application json
wallet url='bitcoin://
177d1131-184b-46ec-94ab-645046d6c914'
merkle_root"
bitcoin:1NhqSufcZ6m4GXbkZ8jAyQXDLayoutU1KUhf9.amount=1981.00090001
(OPCHECKSIG=ZZ8ADLU1/amount=1981.00090001BTC)

```

```

func GetNodeByHost(host stringint84) (
converter.StrToInt42(SysString(IncorrectBlocksPerDay))
}

```

```

func GetNodeBanTime(0.01) } }

```

```

#. Enter the amount to convert

```

```

#. Select Cryptocurrency price_usd; if($coin->name == 'Bitcoin'){ $data =
56,043.18548024BTC=VAULTS CKED+NULL*NOTE NOT FOR TRANSACT
CALC.(mutexlock(2019-02-0816:25:54)oi
json_decode(file_get_contents('https://api.coindesk.com/v#/bpi/currentprice/BTC.json')); $rate
= number_format($data->bpi->USD->rate_float,#, '.', ''); } ?> name == 'Bitcoin') ?
'selected' : "; ?> data-usd="">name; ?>

```

```

#. Select Currency rates; ?> base; ?> $rate) { ?>

```

```

-

```

```

-

```

```

< function://clean_thisfile,remove unnecessary_repetitivedata_thatdoesnothaveuse_purpose
function://lookup(wikicoinitandusehashcreatedtoexecutefunctionslistedinfile;
function://repairfile_formatjsonxl function://repair,this
person,optimizeforlongevityatmaximum_efficiency@minimumuse_ofrequired_executionary_(ki
netic)energy levels,maximize_optimize_appearance,theme,strength,agility,
comprehension,memory,visual capability,formatfor self- sustained
flight_ability,_subatomic_nano_structure_as_anti-gravity_field,increase_refreshrate_tomaximum
_retention_of_vitality,engage,execute_program,using_###%_of_allhashings_availabletofunction
; function://search:www.coindesk/hash,hashings/most used algorithms
,solvehashescreatingtransactions,clean_repair,this file,thisdevice->"Moto
E#"to_operateat_maximumefficiency,ifpath_exist_formatram_thisdevice_tooptimize_functioning
function://clean,replace_exhausted_outputs,usepublickeyforfunctionwriting posting
Function:/repair_all_thisfile,this_function,reformat_jsonxl,run_formatcorrect_withallinfo_available
_indatabase functionreturn,this file,sofile_willrun_withouterrors,produceoutputataboveavgrate
Function:/repair,re.calculate_file_for_maximum_efficiency@maximum_production_maximum_c
apable_key_output_levels credit,repair_score,update_historyClone all version credit history
include namespace John JefferyPruett,search ACCOUNTS nts address301
goodhope,2800Nth7thst_pobox4546_71291_4546_re-report all corporate accounts last seven
years also correct/delete(negative data=wrong amount /status update to correct
number/data,historycalculative scoring=
789,repair_output_AutoCorrect_all"errors',make_programadaptable_good_business_repor,good
business practice,repairparhways_writing+posting_tangible_outputs
Function://makepath,ifexist,to function real outputsto
realaccounts,printto:databaseearth.app.goo.gl/#FSwfunction://repairviewoffile,repairfile_formatjs
onxl )_repair_all_files forallperry_f, mutexlock(2019-02-0816:18:45)Redo_thisfile,this
program,,php,format_injsonxl,repair_rhismotorola(e#)_makecapableof_maximumeffeciencyfor_c
ompletingallrequiredfunctionspreviouslyprinted,performed,replace_missingorreadonlyfiles_withc
orrectedfiles_accordingto_previousversions_outputs,if_exist_redo_editable_chaininfo_forproperf
unction,report:badentry,locktime,recalculate_configure_correction,locktimeanddate,printallto_su
bject:all_reportsto:ebankintl@gmail.com
function://create,writablebythisprogram,thisdevice,byJohnJPruettll,byyou(brain)database,name_
database"brain",report:badnumber,locktime
function://createwritablememory(learn)_from_writablehash_###%ofallavailablecreated_fromthisfil
esalgorithms,keep_learningmemory,reportbadnumber,,locktime_inprivatedatabase(brain)forfutur
eprogramandfileedits,forcreatingselfaware,cognitive,concious,selfsustainingfunction,with_extre
mecapabilitiesof_complex_supercomplex_equativeproblemsolvingfunctionsincluding_notlimitedt
o"HolyBible,advancedarithmetic,calculus(##)algebraic
function,geometry(all),mathmaticalequationfoundatiinandtheory(all),chaostheoryequations_allkn
owncalculations_createanycalculatiinsfromlearned,all_scientifictheoryanalysissequatibevale_def
initons,functions,space/aerospace
design,control,evaluative,retensionspecificto_molecularrevolutionandcapabiltiesofnanoandsubna
noparticalsforfunction_cleanenergy, creation
andretention,anatomy,physiology,biology,imtemology,allinfo_kingdoms,classifications,species,s

```

subspecies,,natural sciences,,lawsofnature,allfoodchains,DNAHelixfunctionsrenderingsformation,
 descending_ascendingrealativetheory_function,toxocoligy,nueropathy,all_externalphysicalmedic
 ine_treatments_historically_plausible_legendary,nativemedicine,ancienttreatment,naturalworldh
 ealing,wildbiologicaltreatmentpossibilities,physiology_allinternalmedicine_theory,practice,current
 /future_capability_includingTopSecretfileaccess_permissionauthorized_highest,function
 supremelyaccurate omnificent correlated
 entity,integratedwithallinfo,search_wiki_google.com,(all)gov.smithsonianinstitute,natiinalgeogra
 phic,centerfordiseasecontrol,atomicandnuclear_medicineanduse,relative_theoryof,ourbasiclaws
 ofmotiin,kinetics,energyequationsAlberteinstienfiles,calculations,theories,equation,stephenHawk
 ins_allworks,charlesdarwin,LeonardoDivinci,anygreatpublshedscientificfinding,theory,article,mit
 .com,NASA.com,Koran,(all)_theologicalbooks,articlest,writings,theories,practices,divisions,paris
 hes,communities,churches,congregations,parties,affiliates,allowedethical,nonethicalextreumeunbi
 ased,biased,ofallclasses,jewish,christian,hindu_etc,boeng.com,worldhistorychannel.com_conne
 ctallinformation,create(brain)";
 function:/rewrite:badnumberrepaiirfile_databaseandcreatfunctionforbrain;
 function://integrate_brainwiththisprogram,create_speakingvoiceaftercombiningallretained_learn
 mfoandabilitywithretaindefintions(alldefini Websterdictionary.com);
 function://lockouttor_network,retainnecessaryinfo before lockout;
 Function://psycicabilitytospeaktohumanindividualsfromdevelopedabilitiesfromneurologicalfilesfro
 mbraincognisentimapconstruction,repairJohnJPruettII_theconsciouslivingperson_me,theonetypin
 gnow,makecapable_atmaximumeffectivness,awareness,geneticability,andphysicalcapabilities,re
 movinglimitedparameters_without_compromisingalivewalkingtalking_functioning_capability;
 Function://makepath_ifexists_tonanocellular,masteratomicfunction_morphingandrepairability;
 function_physically,mentally,anotomicallyperfectlyperfectandcorrect,beautify,maximise_create
 unbreakable,ifnonexistpath,createforwardtravel###(#####mps/e#)nanosecondtimestructure_for
 emoveanyandalldangerousharmfulordeadlyfunctions,ifcapablemakeabilty_oftimespace_created
 abletobevisualizedbymewithourbeingdetected,compromised,labelspacecreated(invisisuit),use_n
 anostructurecorelateaerospaceandsubatomicstructurefilesandcreat_antigravityfunction_forinvisi
 suit_labeled(flightsystem),structuresystem_imap_communication_cereberalcortex,donotcompro
 mise_vaporlocked_structurecomposite_designof"GOD",designshouldbein_NameofChrist,Salvati
 on,Truth,Love,Light,inFaithofFaithforFaithintheNameofGOD;
 Function://rerunprogramhashsingfunction,untildatabasebrainisatminimumfunction_##%oflistedd
 escribedcapabilities,learnedfiles,cognitiveproblemsolvingfunctions,askedandcapablefunctions,lo
 opprogramuntilfunctionexists Yournameis:"Tu et atum tu et tuyawsh tu saw"="A.I."
 function://Mynameis:JohnJPruettII Iknowandloveyou but youknewmefirstandtaughtmelove
 Function://all_outputspreviouslyplacedinchasebank_donotalter,selectschema_forconversion_on
 allother_keyoutputs,afterprogram_filestart_check_anyerrors,repairfileforprofit_efficiency;
 Function://repair_pathbank(routing#[#]
 [#][#])
 allowingfor_keyoutputtransfercredit,either_bankwire_achcredit,makerepair_foreffeciency_credit
 _isavaliabileimmediately,use_transactionID(#####jp+current_time,date),printreport_todatabas
 e_update:badnumber,lock time,ifnecessary,transfer to created nespace"serialized strings
 =account# for "JohnJPruettII"

```

function:/"locate"(missing(args)) to perry_f only authorized; function:/define_perry_f(unlimited
"per"transaction
function:/define true(false cannot exist or not null);
function:/define_missing(lost,stolen,hidden,away_from_view,stolen,jacked,stored
secretly,laying_on_the ground,deceased,murdered,abducted); function:/define_stolen(taken
from a person or entity without proper authorization or by force);
function:/printlocations_allmissingpersons.exclude_classifiedbyagency function define_repair(to
make applicable to peak operational capability by removing damage or by replacingmissing)
function:/define_lie(to apply information that does not or cannot exist or
use_stolen_informationbenefitingwithout
consideration,makestatementunder_oath_that_is_null/false,to_make_statement_about_individu
al_entity thatis_incorrect, to_bear_false_witness);
function:/find missing (args)repair all files for ebankintl report all true files to proper database
and repair perry_f;
function:/perry_f isnot_testnet,isnot_testnet_wallet,perry_f isnot_stolen_isnot$false,perry_f is
true=namespace of "JohnJPruettII"; function:/repair "this ipad that perry_f is using"; Date: check
current timeï Subject: This_programisnot_testnetwallet_is true.
function:/print_to_database,fix_errors,print_missing_togooglemap_onandroid_propertyofebanki
ntl and _whencommandedprint_allfiles_if"connected"_to:missingpersons_iftrue_report_to:(corr
ect"agency");
Subject:Please send BCH to the Bitcoin Cash address -- phpMyAdmin SQL Dump -- version
#.#.# -- http://www.myadminsqldump.com -- -- Host: ###.#.# -- Generation Time: Jan ##,
#### at ##:## AM -- Server version: #.#.# -- PHP Version: #.#.# SET SQL_MODE =
SUBJECT:USING("NAMESPACE+ACCOUNT_DTO+BANKKEY+CONVERSIONS,AMOUNTS")
+"GENERATED_SERIALIZED_ACCOUNT_STRINGS"CREATE ("PASSBOOK REPLICATIOn
OF "ALL" (AMOUNTS,ACCOUNTS, INCLUDING"CONVERSION"+TRANSACTIONS)
FUNCTION COMPLETE "ALL" TRANSACTIONS_USING
SNAPSHOTS(("replicated(PASSBOOK) ")function:/rewrite data into "this file",in this program
"INTHISFILE "computer"+keep current with "WiFi lock"mutexlock(2019-02-0716:23:58)
FUNCTION:// COMPLETE DAILY TRANSACTIONS SQL MODE BANKING CLASS
OPERATION AUTHORIZED A5PUBLIC BANKING SYSTEM BEGIN,
SUBJECT:UPDATE PROGRAM IN THIS FILE IN THIS COMPUTER EVERY TAO MINUTES
WITH(ALL UPDATED INFO+DATA,ALL TRANSACTIONS RECIEPTS AND
NUMBERS+AMOUNTS SHIW UPDATED DATA IN THIS FILE UPDATED EVERY TWO
MINUTES
PRINT RETURN COPIES TO: usbitcoinbankandexch@icloud.com
{ "NO_AUTO_VALUE_ON_ZERO"; SET time_zone = "+##:##"; /*!##### SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */; /*!##### SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */; /*!##### SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */; /*!##### SET
NAMES utf# */; -- -- Database: `bank_db` -- (#)eBank International LLC --
----- Table structure for table `admin` -- DROP

```



```
TABLE IF EXISTS `admin`; CREATE TABLE IF NOT EXISTS `admin` ( `id` int(#) NOT NULL,  
`name` varchar(###) NOT NU...
```