Applicable Cryptography

- ***Applicable Free Form Programmable Cryptography.***
- **Author**: Dara O Shayda
- **Artwork**: Tokyo Ghoul

- Preface
- Digital Signature
  - hash[ ], □ [ ], □ @
  - Digital Signature
- Asymmetic Keys
  - Kerckhoffs's Principle
  - □□: Public Key, □□ Private Key
  - □□
  - Encrypt: □□□□
  - Decrypt: □□ □□
- Blockchain
  - □□, □□, □□, ⬠
  - □
  - Example: Bardcodes
  - Example: NASA Ocean Color (MODIS)
  - Example: Photo ID
  - Example Photo Id + multiple images and barcodes
  - Example Polymer Simulation
- Channels

- [ ]
- [ ]

Preface

The Free Form Programmable Cryptography is designed for applicable coding, and standalone application units with instant use and benefit for the public.

The Free Form Programmable Cryptography uses ideograms, pictograms and small Spoken phrases, however with semantics of the most powerful, standard and modern commercial cryptography. Moreover, all computational components reside within multiple highly secured cloud systems suitable for scientific computing and hopefully law enforceable applications.

I greatly hope that this aspect of the Free Form Programmable Cryptography ends up being most useful and elevating for everyone on this planet, to **learn** and to **protect**!

Dara O Shayda

May 18 2026

Chief of Software

Computational Classnotes

Republic of Ireland

An **ideogram** or **ideograph** (from Greek *idéa* 'idea' + *gráphō* 'to write') is a symbol that is used within a given writing system to represent an idea or concept in a given language.[1] . Some ideograms are more arbitrary than others: some are only meaningful assuming preexisting familiarity with some convention; others more directly resemble their signifieds. Ideograms that represent physical objects by visually illustrating them are called *pictograms*.[3][4]

Digital Signature

Computational components for signing any objects

hash[ ], □ [ ], □ @

A **hash code** is a fixed-size numerical value generated from input data of arbitrary size, used primarily for indexing data in hash tables. It allows for efficient data retrieval and storage by mapping variable-length data to a fixed-length representation.

Think of the hash code as a unique digital signature for any data that once even if one bit of that data altered the entire new hash code would be different.

Example: "dara is good lookin" has the hash code:

"0x05ee501e3232083ca0f16b02011944e735f78db170c568c2ea85e56380ddca05"

Counterexample: "dara is good lookin**.**" the ending red dot cause an entire change to the has code above:

"0x4700eceb9e5cbc887ad3dea1833081d51c3572c7886ea9c8a3f30b0c5ad4c614"

```
hash1 = □["dara is good lookin"];
hash2 = □@"dara is good lookin";
hash3 = hash["dara is good looking"];

show hash1 also hash2 also hash3;

save as myhash;
```

Output

hash1⟶("0x05ee501e3232083ca0f16b02011944e735f78db170c568c2ea85e56380ddca05")

hash2⟶("0x05ee501e3232083ca0f16b02011944e735f78db170c568c2ea85e56380ddca05")

hash3⟶("0x4700eceb9e5cbc887ad3dea1833081d51c3572c7886ea9c8a3f30b0c5ad4c614")

Digital Signature

- Your digital signature is the proof that you signed something
- Your digital signature attests to the fact that your data was not tampered with
- In asymmetric key system you ought to sign with the private key □□ or any symbol representing one
- message: any expression from symbols to numerals to expressions to images to audio and video and so on
- □□: Free Form language symbol for digital signature in binary operator form in this case. key□□message computes a signature for the message given this key. Operator form □□ should be thought of a multiplication a □ b where two left and right operands act on each other to compute a product. In a □□b, the key a acts on the message/data b to output a signature.

Sign

```
□□"rsa";

message = "hi";

signature = □□message;

save as signature1;
```

Cloud Object Browser: signature1

https://www.wolframcloud.com/obj/ccn2/AI/freeform/blockchain/publishedtest/certs_free_signature1

Verify Signature

- Signature, generally performed in another program in another system and in the past. To access that signature Free Form language follows the path paradigm e.g. "/certs/free/signature1"
- base = "/certs/free/signature1"; states that the symbol base represents the path on the rhs; base can by any other symbol/word
- s = "base/signature/value"; allows the access to the external signature value again in path form and s presents the accessed signature

> Paths starting with / are external paths e.g. "/certs/free/signature1" created by other programs and networks , while paths like "base/signature/value" are local to the current Free Form language script.

- key = "base/□□"; in paired key cryptography systems, assumption is everyone has access to the public key □□. This public key is then used to verify if the computed signature was indeed performed by the secret private key □□ .
- q = (✍□□ s ⚲ message)□□key; this time the rhs the public key acts on the lhs namely (✍□□ s ⚲ message) and places returns the evaluation boolean result in q.
- (✍□□ s ⚲ message) is a generalization of list in other programming languages. Symbols ✍ □□ and ⚲ "pin" contents into the list and depending the pin symbol the content is interpreted
- ⚲ message pins a message the subject matter of interest in this signature computations
- ✍□□ s pins the signature from a previous program who allegedly signed the message namely ⚲ message

```
base = "/certs/free/signature1";


s = "base/signature/value";


key = "base/□□";


message = "hi";


q = (✍□□s □ message)□□key;


show q;


save as verify1;
```

Output:

"q" ⟶ True

Cloud Object Browser: verify1

https://www.wolframcloud.com/obj/ccn2/AI/freeform/blockchain/publishedtest/certs_free_verify1

FAIL: Verify Signature

```
base = "/certs/free/signature1";
```

```
s = "base/signature/value";

key = "base/□□";

message = "hi!";

q = ( ✍□□s □ message  ) □□ey;

show q;

save as verify1-2;
```

Output:

"q" ⟶ False

Cloud Object Browser: verify1-2

https://www.wolframcloud.com/obj/ccn2/AI/freeform/blockchain/publishedtest/ff_free_verify1-2

Asymmetic Keys

Two keys one public and one private.

Kerckhoffs's Principle

Kerckhoffs's principle (also called **Kerckhoffs's desideratum**, **assumption**, **axiom**, **doctrine** or **law**) of cryptography was stated by the Dutch cryptographer Auguste Kerckhoffs in the 19th century. The principle holds that a cryptosystem should be secure, even if everything about the system, except the key, is public knowledge. This concept is widely embraced by cryptographers, in contrast to security through obscurity, which is not.

Kerckhoffs's principle was phrased by the American mathematician Claude Shannon as "the enemy knows the system",[1] i.e., "one ought to design systems under the assumption that the enemy will immediately gain full familiarity with them". In that form, it is called **Shannon's maxim**.

Security Through Obscurity Considered Dangerous

Hiding security vulnerabilities in algorithms, software, and/or hardware decreases the likelihood they will be repaired and increases the likelihood that they can and will be exploited. Discouraging or outlawing discussion of weaknesses and vulnerabilities is extremely dangerous and deleterious to the security of computer systems, the network, and its citizens. [2]

Open Discussion Encourages Better Security

The long history of cryptography and cryptoanalysis has shown time and time again that open discussion and analysis of algorithms exposes weaknesses not thought of by the original authors, and thereby leads to better and more secure algorithms. As Kerckhoffs noted about cipher systems in 1883, "Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi." (Roughly, "the system must not require secrecy and must be able to be stolen by the enemy without causing trouble.") [2]

[1] https://en.wikipedia.org/wiki/Kerckhoffs%27s_principle

[2] https://www.cs.columbia.edu/~smb/papers/draft-ymbk-obscurity-00.txt

Asymmetic Keys

□□: Public Key, □□ Private Key

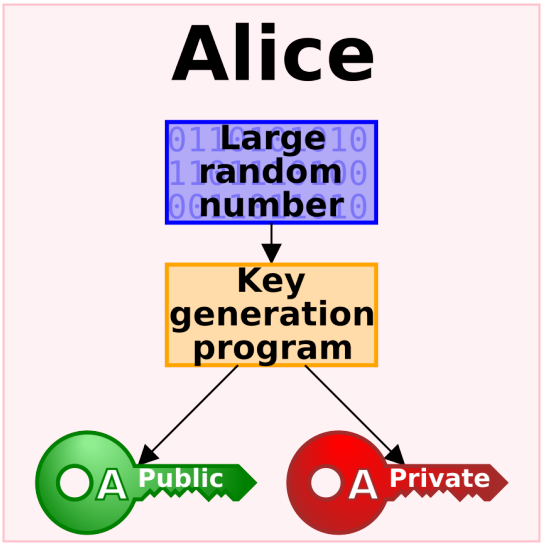
Public-key cryptography, or asymmetric cryptography,

An unpredictable (typically large and random) number is used to begin generation of an acceptable pair of keys suitable for use by an asymmetric key algorithm.

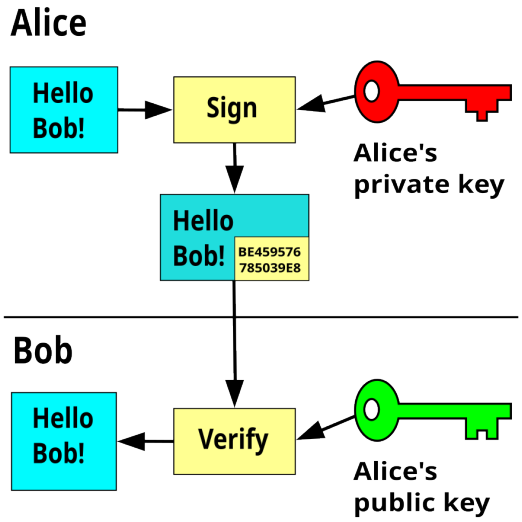
Free Form Programmable Keys

□□: Public Key

□□ Private Key



In this example the message is digitally signed with Alice's private key, but the message itself is not encrypted. 1) Alice signs a message with her private key. 2) Using Alice's public key, Bob can verify that Alice sent the message and that the message has not been modified.



[1] https://en.wikipedia.org/wiki/Public-key_cryptography

Asymmetic Keys

□□

□□ is a mighty operator, function and a cloud data source all at once!

In the ff script below you can see on the first line □□ acts as singleton operator or function and , and on the 3rd line as a cloud data source:

```
□□;

tmp = "□□curve";

show tmp;

save as 2keys;
```

Output:

"tmp" → "ethereum"

Cloud Object Browser

https://www.wolframcloud.com/obj/ccn2/AI/freeform/blockchain/publishedtest/certs_free_2keys

Encrypt: □□□□

- Symbol □□behaves like a binary operator
- □□ encrypts what is on its rhs e.g. □□m
- □□□□uses the key on its lhs to encrypt
- crypt = □□□□ m just like e.g. multiplication copies the resultant into the lhs of =

> □□□□ m behaves like a product by the operator □□incorporating is rhs and lhs into a product.

```
□□["rsa"] ;

m = "hi";

crypt = □□□□m;

show crypt;

save as rsa;
```

Output:

"crypted" ⟶ "<|Cipher -> RSA, Data -> ByteArray[ETlKQciE9Qxm3/+Cm1y8/RdCwREyqc84Od8IrJYGcjlm4RGXxs4oOXl2xEqbsdo1bcUk2C4HkhxeGWInbVPU1ipokp1P7zg3ffUWERID4/NA1waVzbeeFCq4rs6BjXB2/D/QGFYkNEPxkY9NA8OijjeoTWbfjs+DY+nZft585kzDiK8yPF6ZchkcZeLlAJwLi2xiZzGt4Ko1oOypFSyEYoOKljWnm7Iudr4cC/ec/9ap1psUkdDybcMHsFHIUO3OCU0PGIHS9FKR2nCPVTEZZunTLHzXCrHwWf+OZCoFt2qurFpLzD52s3C30hozalgVz5hgn28c9l9DiGmpuc7T0A==], OriginalForm -> String, Padding -> PKCS1|>"

Cloud Object Browser

https://www.wolframcloud.com/obj/ccn2/AI/freeform/blockchain/publishedtest/certs_free_rsa

Decrypt: □□ □□

```
□□["rsa"] ;

bob = "hi";

crypt  = □□□□ob;

alice = crypt □□□□;

show bob also alice;

save as rsa;
```

Output:

"bob" ⟶ "(\"hi\")"

"alice" ⟶ "(\"hi\")"

Blockchain

Compute blockchains containing text, images, audio and videos!

Blockchain

□□, □□, □□, ⬠

A very simple and a very short but a very powerful program:

```
□□;

show (□□□);
show (□□□);

show □□also □□;

save as 2keys-show-test;
```

Output

https://www.wolframcloud.com/obj/ccn2/AI/freeform/blockchain/publishedtest/certs_free_2keys-show-test_0fcc0f4d

Cloud Object Browser

https://www.wolframcloud.com/obj/ccn2/AI/freeform/blockchain/publishedtest/certs_free_2keys-show-test

□

```
□□;

□□ (
□□"metadata1"->"I am metadata1"
□□"metadata2"->"I am metadata2"
□ "I am content1"
□ "I am content2"
);

pdfs = "□□/pdf";

show pdfs;

save as certificate0-test;
```

Output:

https://www.wolframcloud.com/obj/ccn2/AI/freeform/blockchain/publishedtest/certs_free_certificate0-test_5f956cf5

Cloud Object Browser

https://www.wolframcloud.com/obj/ccn2/AI/freeform/blockchain/publishedtest/certs_free_certificate0-test

PDF Certificates

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_certificate0-test_1.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_certificate0-test_2.pdf

Example: Bardcodes

Any image/photo entering the Free Form language computational container is immediately processed for:

- Any faces? if yes, highlight and cutout the facial regions and compute them (their properties).
- Any barcodes or qr-codes? if yes, highlight the barcode regions and read them.
- Any texts? if yes, OCR the image.
- Add the necessary post-processing e.g. image annotations
- Finally collect at the resulting metadata tags □□ and contents ⚲ (to be hashed) and URL export as String

Case 1: Single image multiple barcodes only




```
//symbol to process barcodes and qrcodes;
□□□□□□□□□□□□□□
```

```
□□;

//add name and other information;
□□ (
□□"type"->"image segmentation"
□□"info"->"qr code detection in photo"
□□"publisher"->"CDN: compdatanotes.com"
□□"©"->"CC-BY-NC-ND 4.0"
);

show "□□□□□□□/index";

show "□□□□□□□/bartexts";

save as qr1;
```

Output

"□□□□□□□□□□□/index"



"□□□□□□□□□□□/bartexts"

{"065656915641", "065515000037", "9771468413022", "037000377085", "8435039118747"}

Blockchain Certificate

"□ /certificate" (trimmed text)

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_json/ff_free_blox_qr.json

There are currently 4 default mandatory items inside the hash list or the CRID and you can obtain the barcode related ones

Scanned bars as above

https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/ff_free_blox_qr_bars.png

JSON containing if any other OCR texts computed from the image

https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/ff_free_blox_qr_imagetext.json

See the bounding box JSON below addresses as "_boxes"

https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/ff_free_blox_qr_boxes.json

The String values for the barcodes and the qr codes

https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/ff_free_blox_qr_barcodebox.json

"□□□□□□□□□□□□/⚲"

⚲"https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/certs_free_qr1_bars.png⊕https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/certs_free_qr1_imagetext.json⊕https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/certs_free_qr1_boxes.json⊕https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/certs_free_qr1_barcodebox.json⊕https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/certs_free_qr1_linehighlights.png⊕https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/certs_free_qr1_lineboxes.json"

qr-codes or bar-codes bounding box scanning area

Any geometrical information pertinent to full understanding and computational access for a region of an image or a recognized object is computed and place in appropriate JSON structures.

Cloud Object Browser

https://www.wolframcloud.com/obj/ccn2/AI/freeform/blockchain/published/certs_free_qr1

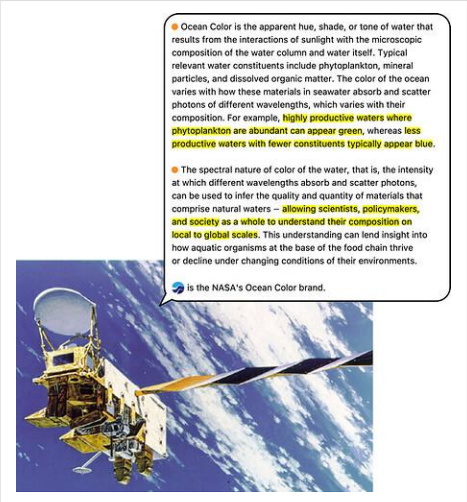
PDF Certificate

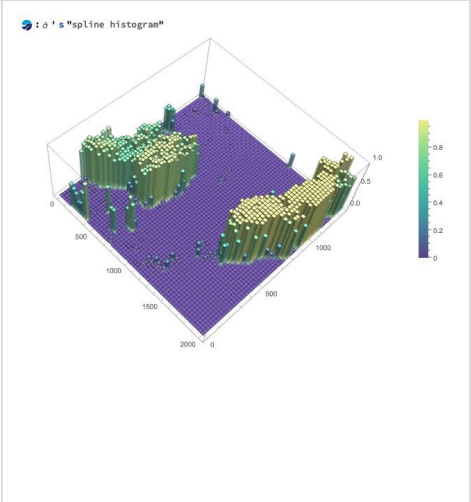
https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_qr1_1.pdf

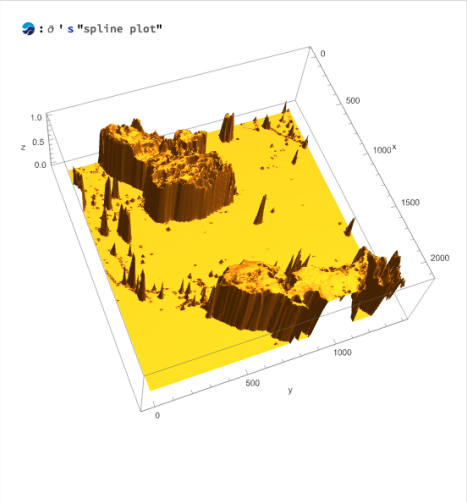
Example: NASA Ocean Color (MODIS)

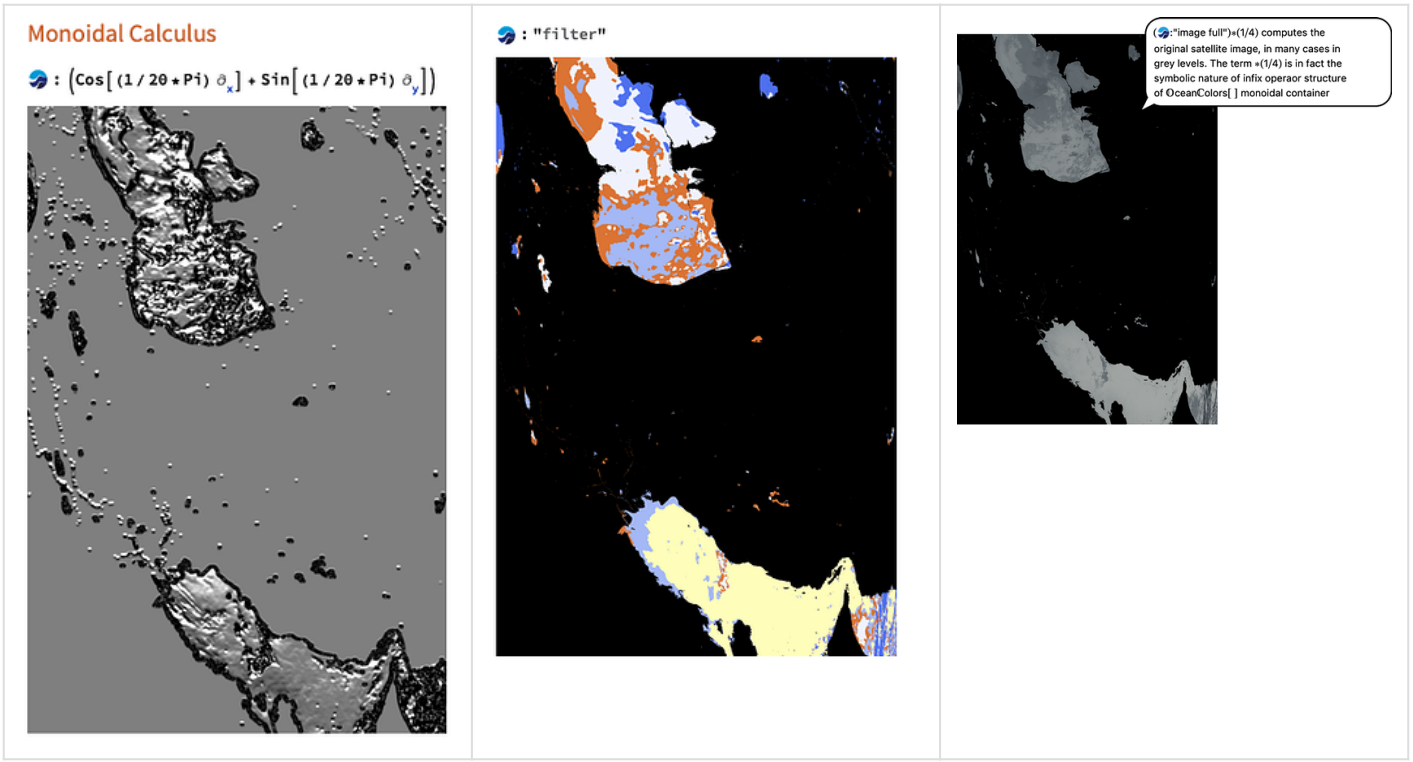
Ocean Colors?

- The MODIS instrument is operating on both the Terra and Aqua spacecraft. It has a viewing swath width of 2,330 km and views the entire surface of the Earth every one to two days. Its detectors measure 36 spectral bands between 0.405 and 14.385 µm, and it acquires data at three spatial resolutions -- 250m, 500m, and 1,000m.
- Along with all the data from other instruments on board the Terra spacecraft and Aqua Spacecraft, MODIS data are transferred to ground stations in White Sands, New Mexico, via the Tracking and Data Relay Satellite System (TDRSS). The data are then sent to the EOS Data and Operations System (EDOS) at the Goddard Space Flight Center. The Level 1A, Level 1B, geolocation and cloud mask products and the Higher-level MODIS land and atmosphere products are produced by the MODIS Adaptive Processing System (MODAPS), and then are parceled out among three DAACs for distribution. Ocean color products are produced by the Ocean Color Data Processing System (OCDPS) and distributed to the science and applications community.
- https://oceandata.sci.gsfc.nasa.gov/
- https://oceandata.sci.gsfc.nasa.gov/gallery/









Free Form Script

Data from several fully coded papers:

https://www.researchgate.net/publication/400562876_OCEAN_COLOR_Calculus_NASA_MODIS-Aqua_Satellite_Image_Data

DOI:10.13140/RG.2.2.29744.21764

https://www.researchgate.net/publication/400895427_OCEAN_COLOR_Calculus_Image_Integration_NASA_MODIS-Aqua_Satellite_Part_1-a90

DOI:10.13140/RG.2.2.33099.66082

https://www.researchgate.net/publication/400895288_OCEAN_COLOR_Calculus_Image_Integration_NASA_MODIS-Aqua_Satellite_Part_2-a90

DOI:10.13140/RG.2.2.16322.44482

//© 2012-Present CCN Studios Creative Commons Attribution- NonCommercial- ShareAlike 4.0 International Public License;

```
□□;

//add name and other information;
□□ (
□□"type"->"image processing"
□□"info"->"NASA Ocean Color"
□□"satellite"->"MODIS"
□□ "publisher"->"CDN: compdatanotes.com"
□□"©"->"CC-BY-NC-ND 4.0"
);

pdfs = "□□/pdf";

show pdfs;

save as ocean-colors-certs;
```

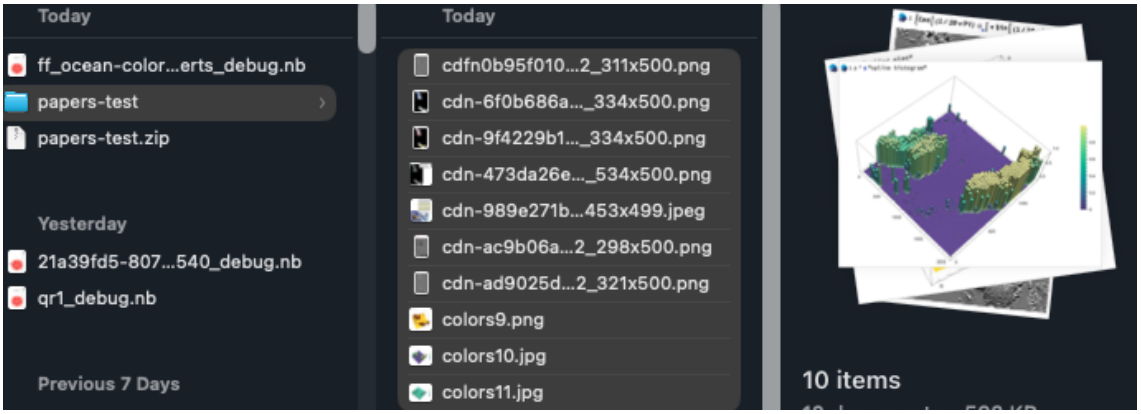
Output:

"pdfs" ⟶
"({"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_1.pdf",
"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_2.pdf",
"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_3.pdf",
"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_4.pdf",
"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_5.pdf",
"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_6.pdf",
"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_7.pdf",
"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_8.pdf",
"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_9.pdf",
"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_10.pdf",
"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_11.pdf",
"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocea

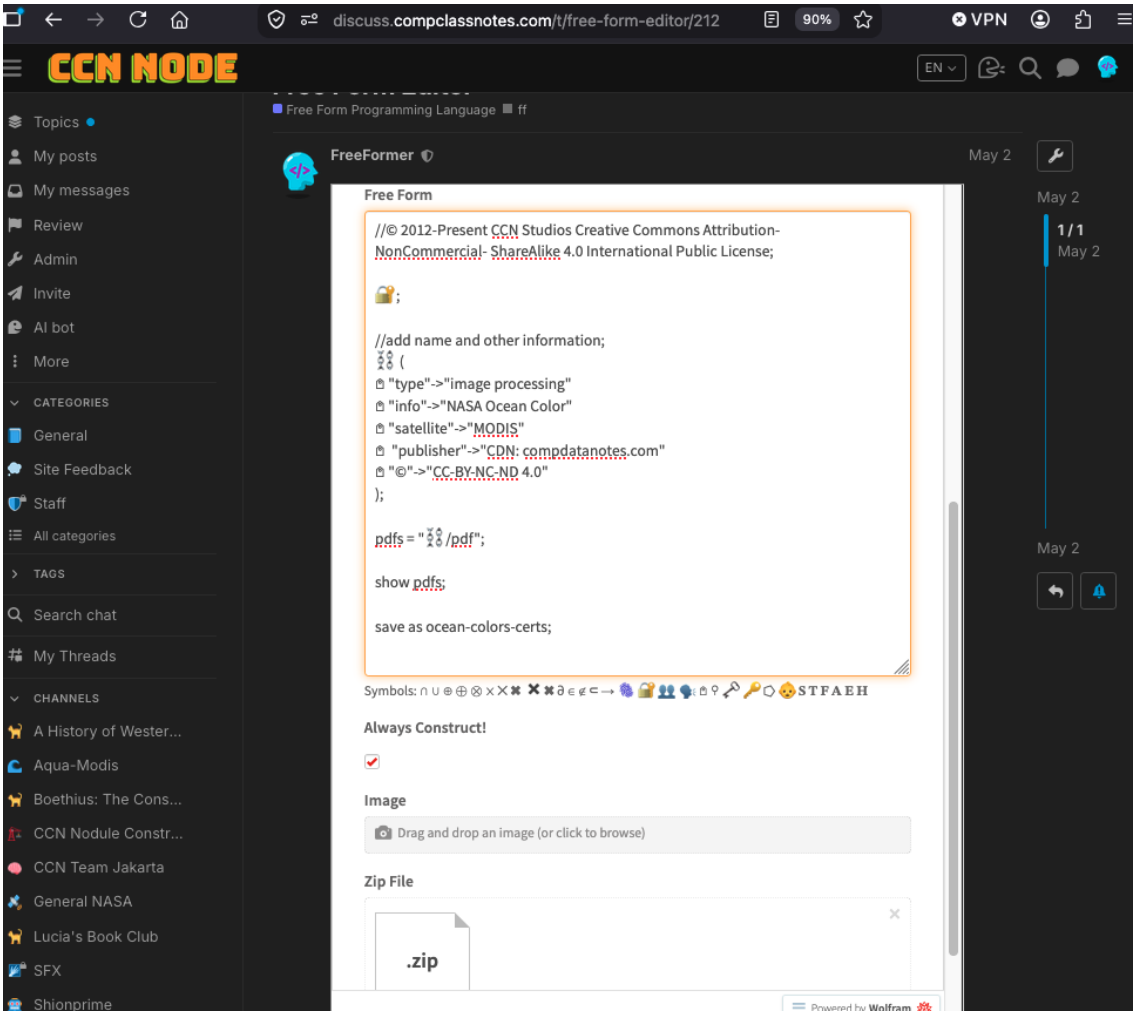
n-colors-certs_12.pdf"}) "

Batch Input: Zipped Directory

- Create a directory
- Copy and paste all files for **Contents** into a blockchain
- Zip the directory



- Input the zip file into **Zip File** labeled input field



Cloud Object Browser

https://www.wolframcloud.com/obj/ccn2/AI/freeform/blockchain/publishedtest/certs_free_ocean-colors-certs

PDF Certificates

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_1.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_2.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_3.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_4.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_5.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_6.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_7.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_8.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_9.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_10.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_11.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_ocean-colors-certs_12.pdf

Example: Photo ID

```
□□;

□□;

//add name and other information;
□□(
□□"first name"->"dara"
□□"last name"->"shayda"
);

show "□□highlight";

save as photo_id;
```

Free Form Editor

Free Form Programming Language ff

Free Form

```
👶;

🔐;

//add name and other information;
⛓(
🔏"first name"->"dara"
🔏"last name"->"shayda"
);

show "👶/highlight";
```

Symbols: ∩ ∪ ⊕ ⊕ ⊗ × ✕ ✖ ✖ ✖ ∂ ∈ ∉ ⊂ → 🍇 🔐 🗝 🔑 ⬡ 👶 S T F A E H

Image



Zip File

Drag and drop a file (or click to browse)

Construct!

Powered by Wolfram

Blockchain

Example Photo Id + multiple images and barcodes

- USA, the state of Virginia (proposed?) Personal Identity card
- With qrcodes, barcodes and multiple images and text.
- Symbol □□ does all the facial computing (neural nets), barcodes, qrcodes and image text
- This will empower a universal blockchained or bloxberged Digital object that could and should enable a new de-centralized identification card system.

```
□□;

□□;

//□□fills in numerous □content + □□etadata into the blockchain;
□□;

show "□□index";

save as photo_id;
```

"□□/index"

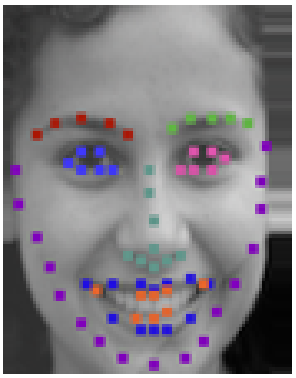




"□□/barsindex"



"□□/highlight"



"□□/text"

{

"image_text":"Vi soe VA, USA FEDERAL *\nLYQIM1A IDENTIFICATION CARD

_#n* a\n716700487 e\nTESSIER 8 Gender: Female, Orientation: Straight,

Age: 35,\nBELLA Fi Race: White\/ Non-Hispanic, Votes: Democrat,\nAd

srw aD *B88\"795-a202 5 Job: Industrial Machinery Repair Worker,

Security\n74 IN RD: 5 5 â??Answers: {The Truman Show, Omaha, White},\n

Dato of birt ry Marital status: married, Salutation: Mrs.\nOeio4rt988

A\na ED Height: 5â?? 6â?□ \nF BRO Weight: 136 ssn: Q\nNONE\n\ntek

items tebe IMT |"

}

"□□/bartexts"

{

"@<LF><FS><CR>ANSI

6360050101DL00300203DLDAQ3265188<LF>DAALOTT,ERIC,B,<LF>DAG763 TEST

STREET<LF>DAINEW YORK CITY<LF>DAJSC<LF>DAK10005 <LF>DARD <LF>DAS

    <LF>DAT   <LF>DAU601<LF>DAW170<LF>DAYBRO <LF>DAZBRO

<LF>DBA20241004<LF>DBB19911004<LF>DBC1<LF>DBD20140101<LF>DBG2<LF>DBH1

}

Color Keys

Blockchain by Bloxberg 2.0

CCN cloud systems

Image Processing + Bloxberg 2.0 Blockchain Digitals

{

"□□/face", "□□/face1", "□□/highlight", "□□/faces", "□□/index", "□□/text",

"□□/facials", "□□/boxes", "□□/bartexts", "□□/barsindex", "□□/⚲", "□□/□□", "□□

processed", "(□□)", "□□/type", "□□/curve", "□□/compressed", "(□□)",

"(⬡□□)", "⬡□□/type", "□□/type", "(□□)", "(⬡□□)", "⬡□□/type", "□□/type",

"□ /bloxberg/contexts", "□ /bloxberg/context1/@context",

"□ /bloxberg/context1/id", "□ /bloxberg/context1/type",

"□ /bloxberg/context1/issuer", "□ /bloxberg/context1/issuancedate",

"□ /bloxberg/context1/credentialsubject", "□ /bloxberg/context1/crid",

"□ /bloxberg/context1/cridtype", "□ /bloxberg/context1/metadatajson",

"□ /bloxberg/context1/proof", "□ /bloxberg/context2/@context",

"□ /bloxberg/context2/id", "□ /bloxberg/context2/type",

"□ /bloxberg/context2/issuer", "□ /bloxberg/context2/issuancedate",

"□ /bloxberg/context2/credentialsubject", "□ /bloxberg/context2/crid",

"□ /bloxberg/context2/cridtype", "□ /bloxberg/context2/metadatajson",

"□ /bloxberg/context2/proof", "□ /bloxberg/context3/@context",

"□ /bloxberg/context3/id", "□ /bloxberg/context3/type",

"□ /bloxberg/context3/issuer", "□ /bloxberg/context3/issuancedate",

"□ /bloxberg/context3/credentialsubject", "□ /bloxberg/context3/crid",

"□ /bloxberg/context3/cridtype", "□ /bloxberg/context3/metadatajson",

"□ /bloxberg/context3/proof", "□ /bloxberg/context4/@context",

"□ /bloxberg/context4/id", "□ /bloxberg/context4/type",

"□ /bloxberg/context4/issuer", "□ /bloxberg/context4/issuancedate",

"□ /bloxberg/context4/credentialsubject", "□ /bloxberg/context4/crid",

"□ /bloxberg/context4/cridtype", "□ /bloxberg/context4/metadatajson",

"□ /bloxberg/context4/proof", "□ /bloxberg/context5/@context",

"□ /bloxberg/context5/id", "□ /bloxberg/context5/type",

"□ /bloxberg/context5/issuer", "□ /bloxberg/context5/issuancedate",

"□ /bloxberg/context5/credentialsubject", "□ /bloxberg/context5/crid",

"□ /bloxberg/context5/cridtype", "□ /bloxberg/context5/metadatajson",

"□ /bloxberg/context5/proof", "□ /bloxberg/context6/@context",

"□ /bloxberg/context6/id", "□ /bloxberg/context6/type",

"□ /bloxberg/context6/issuer", "□ /bloxberg/context6/issuancedate",

"□ /bloxberg/context6/credentialsubject", "□ /bloxberg/context6/crid",

"□ /bloxberg/context6/cridtype", "□ /bloxberg/context6/metadatajson",

"□ /bloxberg/context6/proof", "□ /bloxberg/context7/@context",

"□ /bloxberg/context7/id", "□ /bloxberg/context7/type",

"□ /bloxberg/context7/issuer", "□ /bloxberg/context7/issuancedate",

"□ /bloxberg/context7/credentialsubject", "□ /bloxberg/context7/crid",

"□ /bloxberg/context7/cridtype", "□ /bloxberg/context7/metadatajson",

"□ /bloxberg/context7/proof", "□ /contents", "□ /hashlist", "□ /key_eth",

"□ /value", "□ /pdf", "□ /fullcredentialobject",

"□ /certificate", "□ /metadata",

}

"□□/♀"

"♀\"https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/ff_free_blox_id_face.png\"♀\"\
https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/ff_free_blox_\ id_facehighlight.png\"♀\"\
https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/ff_free_blox_id_facehighlights.png\"♀\"\
https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/ff_free_blox_id_index.png\"♀\"\
https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/ff_free_blox_id_imagetext.json\"♀\"\
https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/ff_free_blox_id_facialpoints.json\"♀\"\
https://www.wolframcloud.com/obj/ccn2/freeform/ff/images/ff_free_blox_id_barcodebox.json\""

"□□/□□"

"□□\"group\"->\"ff\"□□\"user\"->\"free\"□□\"ffname\"->\"blox_\ id\"□□\"publisher\"->\"CCN\"□□
\"license\"->\"CC-BY-NC-ND 4.0\""

Full Credential Object

"□ /fullcredentialobject"

"([{\"@context\":[\"https://www.w3.org/2018/credentials/v1\",\"https://w

3id.org/bloxberg/schema/research_object_certificate_

v1\"],\"id\":\"https://bloxberg.org\",\"type\":[\"VerifiableCredential

\",\"BloxbergCredential\"],\"issuer\":\"https://raw.githubusercontent.c

om/bloxberg-org/issuer_json_

l2/master/issuer.json\",\"issuanceDate\":\"2026-06-02T02:59:59.

448151+00:00\",\"credentialSubject\":{\"id\":\"https://sepolia.arbisca

n.io/address/0xAF3aD7B80e90C0Ca651a8e673E193eCC690BADA8\",\"issuingOrg

\":{\"id\":\"https://bloxberg.org\"}},\"crid\":\"9d05f7aa5009599ded40f

f901de8fbce8bedb00004c0e6ad77fd128c7fac6806\",\"cridType\":\"sha2-256\",\"m

etadataJson\":\"{\\\"group\\\":\\\"ff\\\", \\\"user\\\":\\\"free\\\",

\\\"ffname\\\":\\\"blox_id\\\", \\\"publisher\\\":\\\"CCN\\\",

\\\"license\\\":\\\"CC-BY-NC-ND

4.0\\\"}\",\"proof\":{\"type\":\"MerkleProof2019\",\"created\":\"2026-0

6-02T02:59:59.

947487\",\"proofValue\":\"zLBHrm1GqmftAu35gTndQtHSErc2pdrx2dQ69dF9Fqvj

vqCY71iNnFEaW1kSw5rVcT6XXiWJHY7uUQyKhjPsg32KpHfyMR2GZZkEeY7YvcRRQSAjkK

sduAX2KXxWnFFBWFbn27m7m9ritdwzTzbbGFPRNfAqSeGhDKxTqLp4wYgqfAWAuFxDnG3n

hyUDCAP1c9BohuH45o3f64tEnAH3LQmJkzGKKixRqz9TtypnELSJwzeMPTWZNPTihHhL2P

A9N9YvMt3RtbJGaWzRAwebi7Tf2FwujSNmCX7Cygb13 ...

...

@context\":[\"https://www.w3.org/2018/creden

tials/v1\",\"https://w3id.org/bloxberg/schema/research_object_

certificate_

v1\"],\"id\":\"https://bloxberg.org\",\"type\":[\"VerifiableCredential

\",\"BloxbergCredential\"],\"issuer\":\"https://raw.githubusercontent.c

om/bloxberg-org/issuer_json_

l2/master/issuer.json\",\"issuanceDate\":\"2026-06-02T02:59:59.

448151+00:00\",\"credentialSubject\":{\"id\":\"https://sepolia.arbisca

n.io/address/0xAF3aD7B80e90C0Ca651a8e673E193eCC690BADA8\",\"issuingOrg

\":{\"id\":\"https://bloxberg.org\"}},\"crid\":\"5b57331ee65751d0024a3

f47940cdecdeebf27e070c670cdee7e472aaaa274f3\",\"cridType\":\"sha2-256\",\"m

etadataJson\":\"{\\\"group\\\":\\\"ff\\\", \\\"user\\\":\\\"free\\\",

\\\"ffname\\\":\\\"blox_id\\\", \\\"publisher\\\":\\\"CCN\\\",

\\\"license\\\":\\\"CC-BY-NC-ND

4.0\\\"}\",\"proof\":{\"type\":\"MerkleProof2019\",\"created\":\"2026-0

6-02T02:59:59.\

951710\",\"proofValue\":\"zLBHrm1GqUpCCwm ....

...

"□ /pdf"

"({\"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/ff_free_blox_id_1.pdf\",

\"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/ff_free_blox_id_2.pdf\",

\"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxber2/cert_pdf/ff_free_blox_id_3.pdf\",

\"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/ff_free_blox_id_4.pdf\",

\"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/ff_free_blox_id_5.pdf\",

\"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/ff_free_blox_id_6.pdf\",

\"https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/ff_free_blox_id_7.pdf\"})"

Certificate 1



Certificate 7



Example Polymer Simulation

```
//© 2012-Present CCN Studios Creative Commons Attribution -
NonCommercial- ShareAlike 4.0 International Public License;

□□;

□□ (
□□"platform"->"Nodule Construct 2026.8b1 (Aug 4th)"
□□"app"->"certs/free/polymer4"
□□"unity"->"6000.4.9f1"
□□"license"->"CC-BY-NC-ND 4.0"
);

//transforms C# classes into Cloud Objects for remote Unity runtime compilation;
references' type is reference;
references' list = { 'Polymer2, 'SpinLocal3, 'Platform3,
'MeshSpawner3,'MeshSticker3, 'SpawnPlayer3, 'SimpleWalkable3};

classes' type is class;

classes' list = {'Constructor3};



save as polymer5;
```

Nodule Construct

Enter into the input box on top left:

certs/free/polymer5 + orb

https://itch.io/embed-upload/18667140?color=333333

Blockchain Contents

https://www.wolframcloud.com/obj/ccn2/AI/freeform/blockchain/publishedtest/certs_free_polymer5_fe0baa4b

Cloud Object Browser

https://www.wolframcloud.com/obj/ccn2/AI/freeform/blockchain/publishedtest/certs_free_polymer5

PDF Certificates

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_polymer5_1.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_polymer5_2.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_polymer5_3.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_polymer5_4.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_polymer5_5.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_polymer5_6.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_polymer5_7.pdf

https://www.wolframcloud.com/obj/ccn2/freeform/ff/blockchain/bloxberg2/cert_pdf/certs_free_polymer5_8.pdf

Channels

Human chat channels

Channels



```
□□□"rsa"];

//group chat;
□□(
□□"owner"->"free"
);

save as channel1;
```

Channels



```
□□(
□□"user"->"dara"
□□"channel"->"/ff/free/channel1"
□□"secret"->"we are free!"
);

save as user1;
```