# #

Bech32 on IRIShub

Bech32 is a new Bitcoin address format proposed by Pieter Wuille and Greg Maxwell. Besides Bitcoin addresses, Bech32 can encode any short binary data. In the IRIS network, keys and addresses may refer to a number of different roles in the network like accounts, validators etc. The IRIS network is designed to use the Bech32 address format to provide robust integrity checks on data. The human readable part(HRP) makes it more efficient to read and the users could see error messages. More details inbip-0173open in new window

#

Human Readable Part Table

HRP Definition iaa IRIShub Account Address iap IRIShub Account Public Key iva IRIShub Validator's Operator Address ivp IRIShub Validator's Operator Public Key ica Tendermint Consensus Address icp Tendermint Consensus Public Key

#

Encoding

Not all interfaces to IRIShub users should be exposed as bech32 interfaces. Many addresses are still in hex or base64 encoded form.

To covert between other binary representation of addresses and keys, it is important to first apply the Amino encoding process before bech32 encoding.

#

Account Key Example

Account Key, aka.Application Key . Once you create a new address, you should see the following:

NAME: TYPE: ADDRESS: PUBKEY: test1local iaa18ekc4dswwrh2a6lfyev4tr25h5y76jkpclyxkz iap1addwnpepqgxa40ww28uy9q46gg48g6ulqdzwupyjcwfumgfjpvz7krmg5mrnwk5xq9l This means you have created a new addressiaa18ekc4dswwrh2a6lfyev4tr25h5y76jkpclyxkz , with the HRPiaa . And its public key could be encoded intoiap1addwnpepqgxa40ww28uy9q46gg48g6ulqdzwupyjcwfumgfjpvz7krmg5mrnwk5xq9l , with the HRPiap .

#

Validator Key Example

Validator Key, aka.Tendermint Key . A Tendermint Consensus Public key is generated when the node is created withiris init . You can get this value with

iris tendermint show-validator--home

< iris-home> Example Output:

icp1zcjduepqzuz420weqehs3mq0qny54umfk5r78yup6twtdt7mxafrprms5zqsjeuxvx