```
BIP: 383
Layer: Applications
Title: Multisig Output Script Descriptors
Author: Pieter Wuille <pieter@wuille.net>
        Andrew Chow <andrew@achow101.com>
Comments-Summary: No comments yet.
Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0383
Status: Draft
Type: Informational
Created: 2021-06-27
License: BSD-2-Clause
```

Abstract

This document specifies `multi()`, and `sortedmulti()` output script descriptors. Both functions take a threshold and one or more public keys and produce a multisig output script. `multi()` specifies the public keys in the output script in the order given in the descriptor while `sortedmulti()` sorts the public keys lexicographically when the output script is produced.

Copyright

Motivation

The most common complex script used in Bitcoin is a threshold multisig. These expressions allow specifying multisig scripts as a descriptor.

Specification

Two new script expressions are defined: `multi()`, and `sortedmulti()`. Both expressions produce the scripts of the same template and take the same arguments. They are written as `multi(k,KEY_1,KEY_2,...,KEY_n)`. `k` is the threshold - the number of keys that must sign the input for the script to be valid. `KEY_1,KEY_2,...,KEY_n` are the key expressions for the multisig. `k` must be less than or equal to `n`.

`multi()` and `sortedmulti()` expressions can be used as a top level expression, or inside of either a `sh()` or `wsh()` descriptor. Depending on the higher level descriptors, there may be restrictions on the type of public keys that can be included.

Depending on the higher level descriptors, there are also restrictions on the number of keys that can be present, i.e. the maximum value of `n`. When used at the top level, there can only be at most 3 keys. When used inside of a `sh()`

expression, there can only be most 15 compressed public keys (this is limited by the P2SH script limit). Otherwise the maximum number of keys is 20.

The output script produced also depends on the value of `k`. If `k` is less than or equal to 16:

```
OP_k KEY_1 KEY_2 ... KEY_n OP_CHECKMULTISIG
```

if `k` is greater than 16:

```
k KEY_1 KEY_2 ... KEY_n OP_CHECKMULTISIG
```

`sortedmulti()`

The only change for `sortedmulti()` is that the keys are sorted lexicographically prior to the creation of the output script. This sorting is on the keys that are to be put into the output script, i.e. after all extended keys are derived.

Multiple Extended Keys

When one or more the key expressions in a `multi()` or `sortedmulti()` expression are extended keys, the derived keys use the same child index. This changes the keys in lockstep and allows for output scripts to be indexed in the same way that the derived keys are indexed.

Test Vectors

Valid descriptors followed by the scripts they produce. Descriptors involving derived child keys will have the 0th, 1st, and 2nd scripts listed.

- `multi(1,L4rK1yDtCWekvXuE6oXD9jCYfFNV2cWRpVuPLBcCU2z8TrisoyY1,5KYZdUEo39z3FPrtuX2QbbwGnN`
  - `512103a34b99f22c790c4e36b2b3c2c35a36db06226e41c692fc82b8b56ac1c540c5bd4104a34b99f22`
- `multi(1,03a34b99f22c790c4e36b2b3c2c35a36db06226e41c692fc82b8b56ac1c540c5bd,04a34b99f22c`
  - `512103a34b99f22c790c4e36b2b3c2c35a36db06226e41c692fc82b8b56ac1c540c5bd4104a34b99f22`
- `sortedmulti(1,04a34b99f22c790c4e36b2b3c2c35a36db06226e41c692fc82b8b56ac1c540c5bd5b8dec5`
  - `512103a34b99f22c790c4e36b2b3c2c35a36db06226e41c692fc82b8b56ac1c540c5bd4104a34b99f22`
- `sh(multi(2,[00000000/111'/222]xprvA1RpRA33e1JQ7ifknakTFpgNXPmW2YvmhqLQYMmrj4xJXXWYpDPS3`
  - `a91445a9a622a8b0a1269944be477640eedc447bbd8487`
- `sortedmulti(2,xpub6ERApfZwUNrhLCkDtcHTcxd75RbzS1ed54G1LkBUHQVHQKqhMkhgbmJbZRkrgZw4koxb5`
  - `5221025d5fc65ebb8d44a5274b53bac21ff8307fec2334a32df05553459f8b1f7fe1b62102fbd47cc80`
  - `52210264fd4d1f5dea8ded94c61e9641309349b62f27fbffe807291f664e286bfbe6472103f4ece6dfc`
  - `5221022ccabda84c30bad578b13c89eb3b9544ce149787e5b538175b1d1ba259cbb83321024d902e1a2`
- `wsh(multi(2,xprv9s21ZrQH143K31xYSDQpPDxsXRTUcvj2iNHm5NUtrGiGG5e2DtALGdso3pGz6ssrdK4PFmM`
  - `0020b92623201f3bb7c3771d45b2ad1d0351ea8fbf8cfe0a0e570264e1075fa1948f`
  - `002036a08bbe4923af41cf4316817c93b8d37e2f635dd25cfff06bd50df6ae7ea203`
  - `0020a96e7ab4607ca6b261bfe3245ffda9c746b28d3f59e83d34820ec0e2b36c139c`
- `sh(wsh(multi(16,03669b8afcec803a0d323e9a17f3ea8e68e8abe5a278020a929adbec52421adbd0,0260`
  - `a9147fc63e13dc25e8a95a3cee3d9a714ac3afd96f1e87`
- `wsh(multi(20,KzoAz5CanayRKex3fSLQ2BwJpN7U52gZvxMyk78nDMHuqrUxuSJy,KwGNz6YCCQtYvFzMtrC6D`

    - 0020376bd8344b8b6ebe504ff85ef743eaa1aa9272178223bcb6887e9378efb341ac
- sh(wsh(multi(20,KzoAz5CanayRKex3fSLQ2BwJpN7U52gZvxMyk78nDMHuqrUxuSJy,KwGNz6YCCQtYvFzMtr
    - a914c2c9c510e9d7f92fd6131e94803a8d34a8ef675e87

Invalid descriptors

- More than 15 keys in P2SH multisig: sh(multi(16,03669b8afcec803a0d323e9a17f3ea8e68e8abe5a2780
- Invalid threshold: multi(a,03a34b99f22c790c4e36b2b3c2c35a36db06226e41c692fc82b8b56ac1c540c5
- Threshold of 0: multi(0,03a34b99f22c790c4e36b2b3c2c35a36db06226e41c692fc82b8b56ac1c540c5bd
- Threshold larger than keys: multi(3,L4rK1yDtCWekvXuE6oXD9jCYfFNV2cWRpVuPLBcCU2z8TrisoyY1,5KY

Backwards Compatibility

multi(), and sortedmulti() descriptors use the format and general operation specified in 380. As these are a wholly new descriptors, they are not compatible with any implementation. However the scripts produced are standard scripts so existing software are likely to be familiar with them.

Reference Implementation

multi(), and sortedmulti() descriptors have been implemented in Bitcoin Core since version 0.17.